# AlgoMesh Training Workshop

*Schedule, Slides and Tutorial Notes*

21 September 2015

# AlgoMesh Training Workshop Schedule: 21 September 2015

| TIME | DESCRIPTION |
| --- | --- |
| **9:00 am** | **Session 1 (2 hr 20 min)**<br>**Introduction and Mesh Generation**<br>• **Slides (1a)**: Introduction to AlgoMesh<br>• **Demo**: Mesh generation overview<br>• **Slides (1b)**: Mesh generation algorithms<br>• **Demo**: Detailed mesh generation<br>• **Tutorial 1**: Mesh generation |
| **11:20 am** | *Morning tea (20 min)* |
| **11:40 am** | **Session 2 (1 hr 20 min)**<br>**MODFLOW-USG Model Building (Steady-State, Part 1)**<br>• **Slides (2a)**: Building multi-layered MODFLOW-USG models<br>• **Demo**: Multi-layered model build<br>• **Slides (2b)**: Cutting out unnecessary model cells<br>• **Demo**: Restricted cell count build |
| **1:00 pm** | *Lunch (1 hr)* |
| **2:00 pm** | **Session 3 (1 hr 20 min)**<br>**MODFLOW-USG Model Building (Steady-State, Part 2)**<br>• **Slides (3)**: Assigning properties and boundary conditions<br>• **Demo**: Assign properties and boundary conditions, run model, import heads<br>• **Tutorial 2**: Steady-state model building |
| **3:20 pm** | *Afternoon tea (20 min)* |
| **3:40 pm** | **Session 4 (1 hr 20 min)**<br>**MODFLOW-USG Model Building (Transient) and Conclusion**<br>• **Slides (4a)**: Transient model setup<br>• **Demo**: Convert steady-state model to transient<br>• **Tutorial 3**: Transient model building<br>• **Slides (4b)**: Conclusion |
| **5:00 pm** | *End of workshop* |

# Introduction to AlgoMesh

SESSION 1: INTRODUCTION AND MESH GENERATION

HYDRO ALGORITHMICS

## Overview of Workshop

- Four sessions, comprising seven topics
  1. INTRODUCTION AND MESH GENERATION … [2:20]
     a) Introduction
     b) Mesh Generation Algorithms
  2. MODFLOW-USG MODEL BUILDING (STEADY-STATE, PART 1) … [1:20]
     a) Building Multi-layered MODFLOW-USG Models
     b) Cutting Out Unnecessary Model Cells
  3. MODFLOW-USG MODEL BUILDING (STEADY-STATE, PART 2) … [1:20]
     -) Assigning Properties and Boundary Conditions
  4. MODFLOW-USG MODEL BUILDING (TRANSIENT) AND CONCLUSION … [1:20]
     a) Transient Model Setup
     b) Conclusion

# Overview of Workshop

- Three modes of training
  - Slides
  - Software demonstration
  - Practical tutorials
- Software demonstration will illustrate most of the techniques needed for the practical tutorials
  - Follow on your own laptop during demos if desired
  - Tutorial notes contain all the steps to follow if you miss anything
- AlgoMesh version provided will run for 30 days
  - Can continue with tutorial material later if you don't get through all of it
  - Email me with questions: damian.merrick@hydroalgorithmics.com

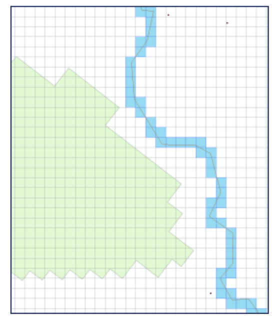


# Workshop USB

- There are four folders on the provided USB key
  - **workshop**
    - Presentation slides in PDF format
    - Tutorial notes in PDF format
    - Tut1, Tut2 and Tut3 folders to go with each of the three tutorials
    - Copy this folder across first!
  - **completed-workshop**
    - Complete MODFLOW-USG model files produced at the end of tutorials 2 and 3
    - Only needed if you want to experiment with model results before completing the tutorials
  - **gwutils**
    - John Doherty's *Groundwater Data Utilities* from http://pesthomepage.org/
    - See gwutils\doc\gwutils_c.pdf for details of the USG-related utilities
    - Should not need this for tutorials, as the one utility used (USGMOD2SMP) is included in Tut3\Working
  - **modflow-usg**
    - USGS' complete MODFLOW-USG version 1.2 release from http://water.usgs.gov/ogw/mfusg/
    - MODFLOW-USG report and input/output file format documentation is useful; see doc folder
    - Should not need this for tutorials, as USG executables are included in Tut2\Working and Tut3\Working

# Benefits of Unstructured Grids

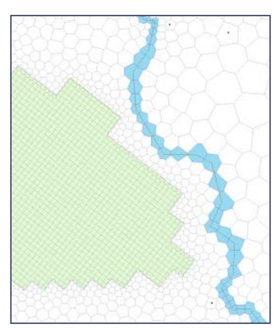
- Traditional finite-difference methods use structured rectangular grids
  - **Easy** to set up
  - All cells **same size** (at least all in a row/column)
  - All cells aligned in the **same direction**





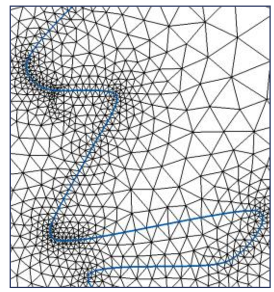
# Benefits of Unstructured Grids

- Control-volume finite-difference (CVFD) formulation of MODFLOW-USG allows unstructured grids to be used
  - Different cell **shapes** (triangles, quadrilaterals, arbitrary polygons)
  - Varying **sizes**
  - Varying **orientations**



# Benefits of Unstructured Grids

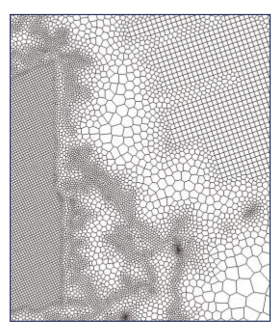
- Unstructured grids give significant benefits:
  - **Closely follow** geographical / geological features
  - **Refine** more heavily in important model areas, **reduce cell counts** in less critical areas
  - **Align cells** in preferential flow directions





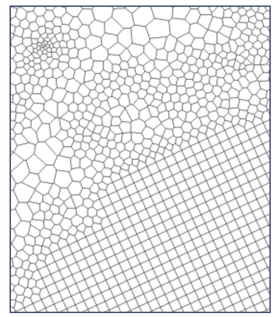
# Benefits of Unstructured Grids

- Unstructured grids give significant benefits:
  - **Closely follow** geographical / geological features
  - **Refine** more heavily in important model areas, **reduce cell counts** in less critical areas
  - **Align cells** in preferential flow directions



# Benefits of Unstructured Grids

- Unstructured grids give significant benefits:
  - **Closely follow** geographical / geological features
  - **Refine** more heavily in important model areas, **reduce cell counts** in less critical areas
  - **Align cells** in preferential flow directions





# Benefits of Unstructured Grids

- BUT: It is **more difficult** to create appropriate unstructured grids
- **Need tools** to assist with grid generation

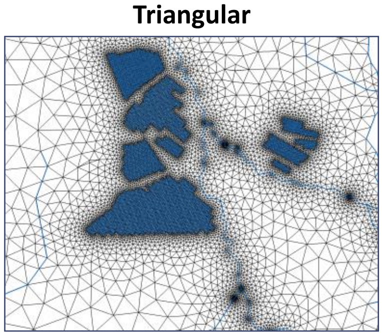
# The AlgoMesh Software

- Developed and progressively improved over the past 2 years
- Initial focus: Groundwater flow simulation with **MODFLOW-USG**
- Can also be used with other simulation models:
  - **HydroGeoSphere**™ (2D mesh interchange in AH2 format)
  - **FEFLOW**® (ASCII supermesh import, 2D mesh interchange via SHP)
  - Anything that can take in a **polygon shapefile** or similar GIS format

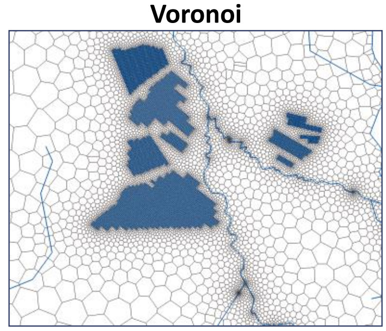


# Types of Mesh

**Triangular**

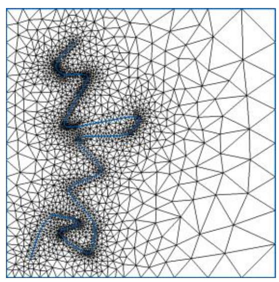


**Voronoi**

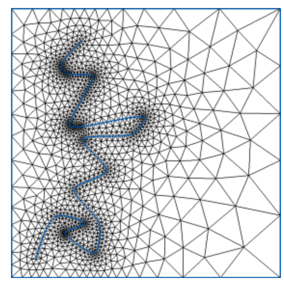


# Mesh Generation Methods

**Delaunay Refinement**
Very fast, reasonable quality and cell count
*Below: 4081 triangles / 0.06 seconds*



**Multi-level Optimization**
Slower, very high quality, lower cell count
*Below: 2941 triangles / 4 minutes 9 seconds*



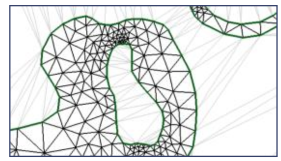


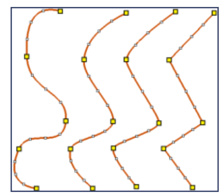
# Mesh Generation Methods

- Delaunay refinement
  - Similar approach to the popular software *Triangle* (but parallelised)
  - See e.g. review of algorithms in Cheng et al., 2012
    Siu-Wing Cheng, Tamal Krishna Dey, and Jonathan Richard Shewchuk, Delaunay Mesh Generation, xii+375 pages, CRC Press, Boca Raton, Florida, 2012.
- Multi-level optimization (variational method)
  - Interleaved Delaunay refinement and weighted Lloyd relaxation approach
  - Tournois, Jane, Pierre Alliez, and Olivier Devillers, Interleaving Delaunay Refinement and Optimization for 2D Triangle Mesh Generation, in Proc. 16th International Meshing Roundtable, Springer Berlin Heidelberg, 2008.

# Graphical User Interface Capabilities

- **Import model geometry** from GIS or manually digitise
  - Points, polylines, polygons
  - Cut out unneeded areas of model
- **Resample polylines** easily to control cell sizing
  - May be variable along length of polyline (point spacing in geometric progression)
- **Fit spline curves** to polylines to represent smoothly curving features

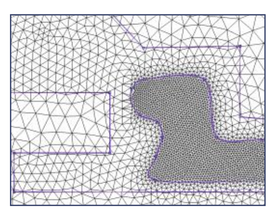


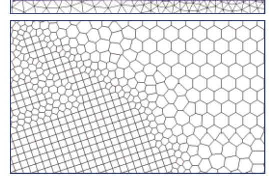




# Graphical User Interface Capabilities

- Specify polygonal regions of **varying cell sizes** (may be nested)
- **Map values from TINs** (interpolated) or polygons onto cells
  - Material properties
  - Boundary conditions
- OR **directly import** property values from CSV (full control over parametrization)
- Insert **structured sub-grids** to improve accuracy in important areas
  - Aligned rectangles or perfect hexagons (in Voronoi meshes)
  - Equilateral triangles (in triangular meshes)





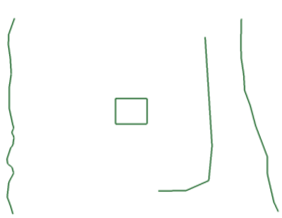
# Graphical User Interface Capabilities

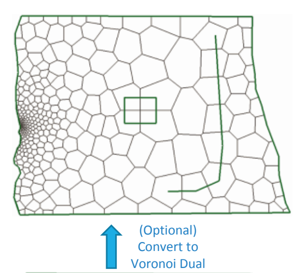
- Creates **MODFLOW-USG model inputs** and related files
  - Base inputs (BAS, NAM, OC, SMS)
  - Unstructured discretization (DISU)
  - Layer property flow (LPF)
  - Time-variant materials (TVM)
  - Specified head boundaries (CHD)
  - General head boundaries (GHB)
  - Pumping wells (WEL)
  - Evapotranspiration (EVT)
  - Recharge (RCH)
  - Drains (DRN)
  - Rivers (RIV)
  - 3D model for ParaView visualization (VTK)
  - Grid specification file (GSF) for interoperation with other MODFLOW-USG utilities

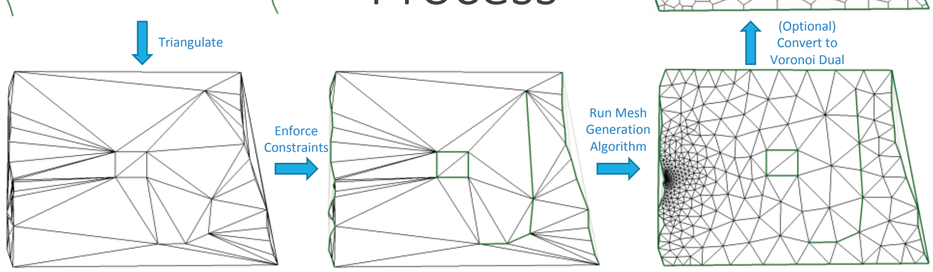
# Mesh Generation Algorithms

SESSION 1: INTRODUCTION AND MESH GENERATION

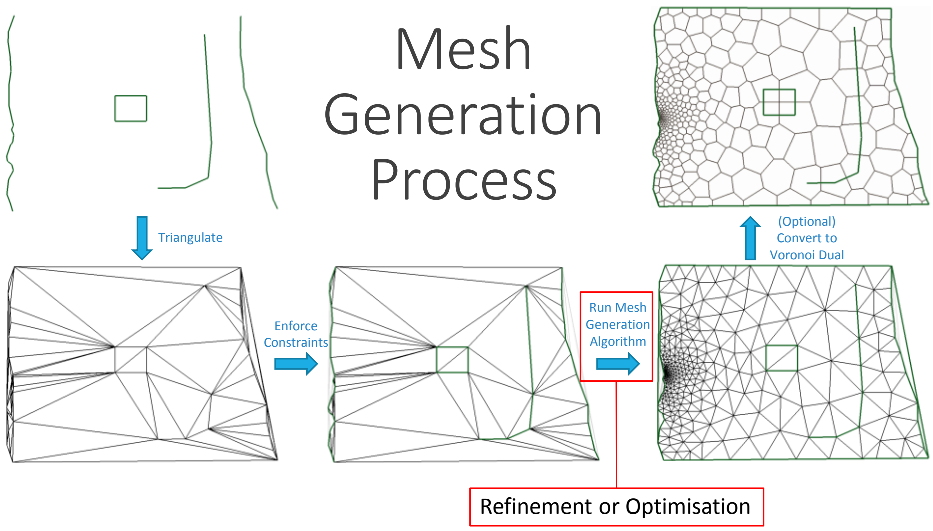
# Mesh Generation Process





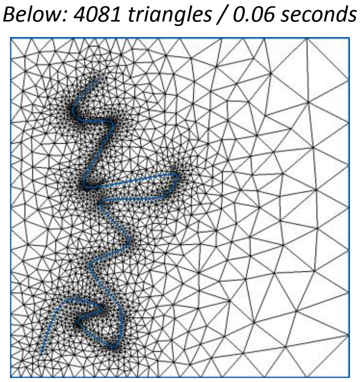



# Mesh Generation Process

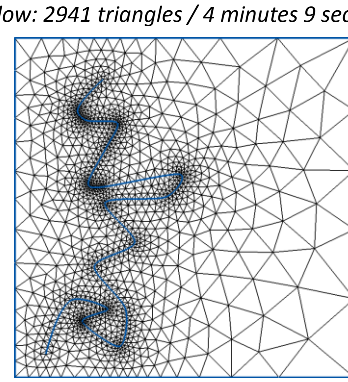



# Mesh Generation Methods

**Delaunay Refinement**
Very fast, reasonable quality and cell count
*Below: 4081 triangles / 0.06 seconds*

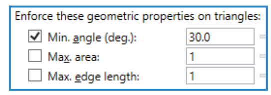


**Multi-level Optimisation**
Slower, very high quality, lower cell count
*Below: 2941 triangles / 4 minutes 9 seconds*

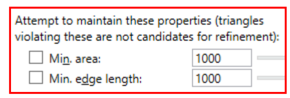


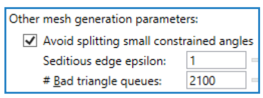
# Delaunay Refinement

- Identify “bad” triangles:
  - Angle at vertex too small (< min angle)
  - Too large (> max area or > max edge length)




- With some exclusions:
  - Too small (< min area or < min edge length) *- avoid this!*
  - Seditious angles *(no reason to disable this)*

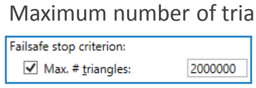


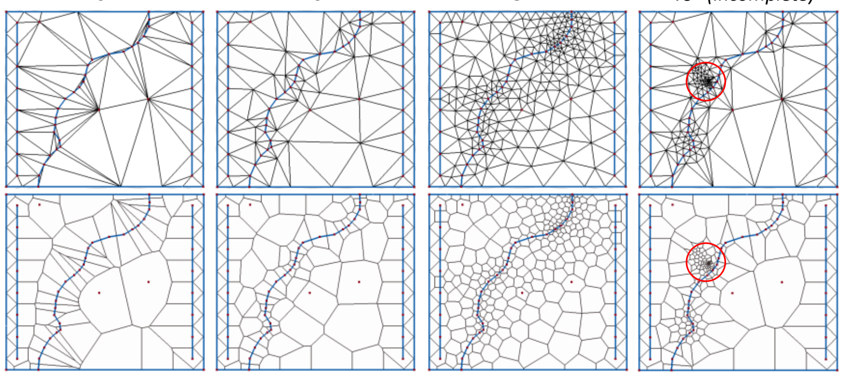



# Delaunay Refinement Algorithm

- Maintain a queue of bad triangles
  - Sorted by shortest edge length
- For each bad triangle:
  - Insert a new point at (or near) the triangle’s circumcentre
  - Adjust the triangulation to include the new point
  - Reassess altered triangles for “badness”
- Stop when:
  - No more bad triangles; OR
  - Maximum number of triangles reached

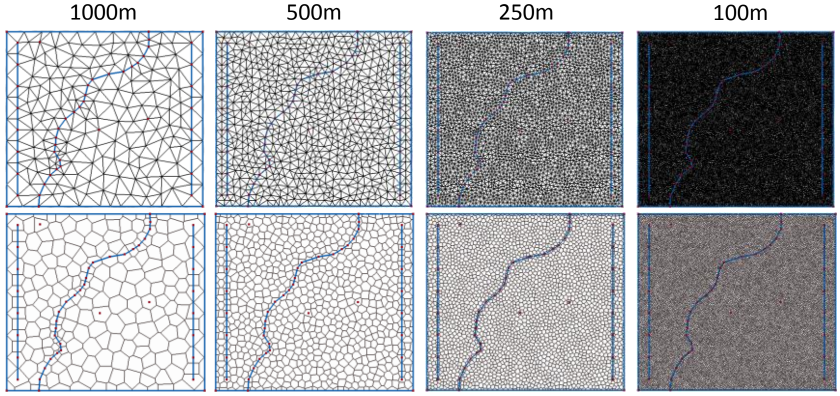


# Delaunay Refinement: Minimum Angle



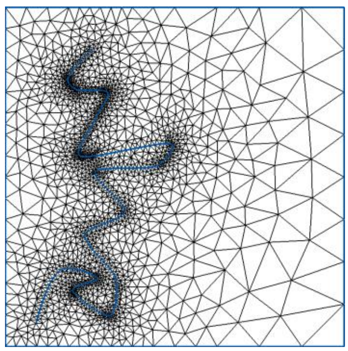



# Delaunay Refinement: Maximum Edge Length

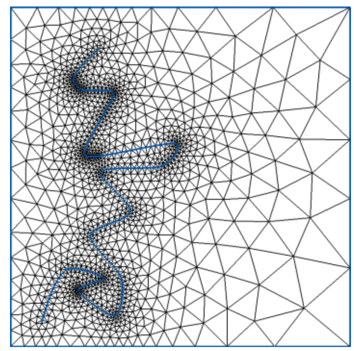


# Mesh Generation Methods

**Delaunay Refinement**
Very fast, reasonable quality and cell count
*Below: 4081 triangles / 0.06 seconds*



**Multi-level Optimisation**
Slower, very high quality, lower cell count
*Below: 2941 triangles / 4 minutes 9 seconds*



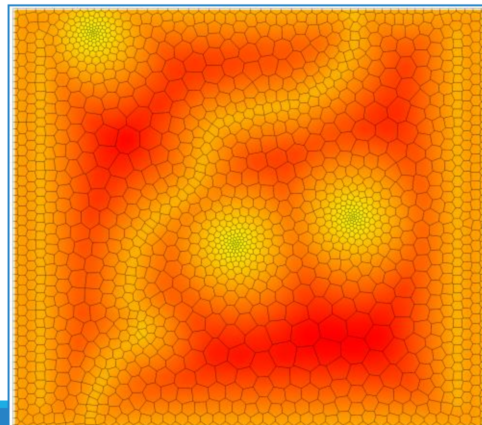


# Optimisation

- Actually a multi-level **variational** approach
  - Iteratively minimise an energy functional over the mesh
- Overall idea:
  - Add a few points (first level)
  - Smooth mesh (variational method)
  - Add a few more points (second level)
  - …

# Optimisation:
# Edge Length Field Generation

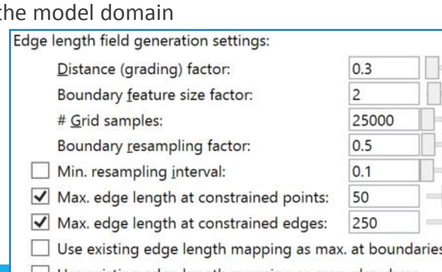
- The energy functional is derived from an **edge length field (ELF)**
  - Specifies the desired cell size at any point in the model domain





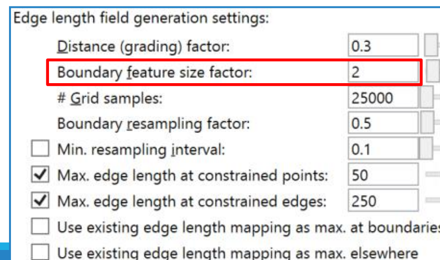
# Optimisation:
# Edge Length Field Generation

- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries



# Optimisation: Edge Length Field Generation

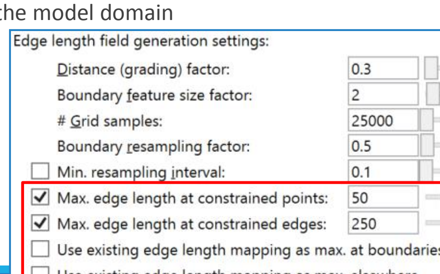
- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries





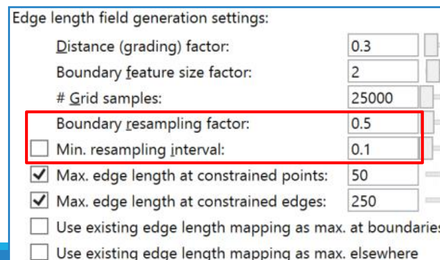

# Optimisation: Edge Length Field Generation

- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries



# Optimisation: Edge Length Field Generation

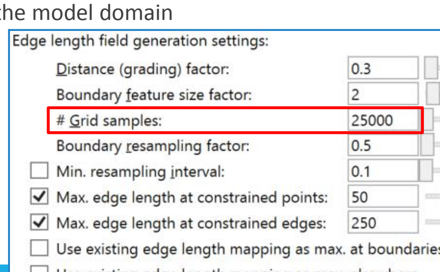
- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries





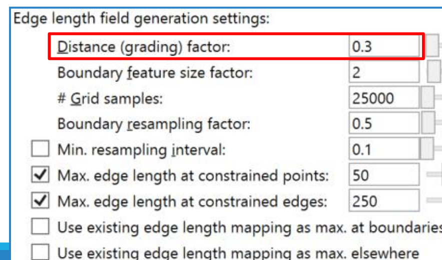

# Optimisation: Edge Length Field Generation

- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries



# Optimisation: Edge Length Field Generation

- AlgoMesh includes automated edge length field generation (ELFG)
  - Takes as input: point and line geometry, cell size constraints, parameters
- The computed ELF is an approximation to local feature size (LFS)
  - Cell size at boundaries depends on proximity to other boundaries (and is limited by user-specified cell size constraints)
  - Constructed as a TIN comprising resampled points along boundaries and points in a regular grid over the model domain
  - Cell size gradually increases away from boundaries






# Optimisation Algorithm

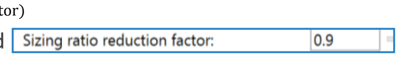
1. Calculate max sizing ratio over all edge constraints and triangles:

$$\text{Sizing ratio} = \frac{\text{Current edge length}}{\text{Desired edge length from ELF}}$$

2. Calculate initial target sizing ratio:

$$\text{Target sizing ratio} = (\text{max sizing ratio}) \times (\text{reduction factor})$$

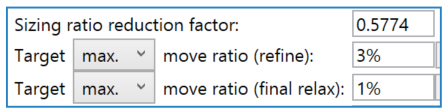
   - Reduction factor (< 1.0) is user-specified




3. Refine edge constraints and triangles with sizing ratio larger than the target sizing ratio
   - Add point at edge midpoint or triangle circumcentre
4. Move all points to optimal positions using weighted Lloyd relaxation
   - Iterative process (keep moving points until amount of movement is small)
   - Compute approximate integral (using quadratures) of desired edge length from ELF over each point's Voronoi cell, then move the point to its new locally minimal weight Voronoi centre, and repeat…

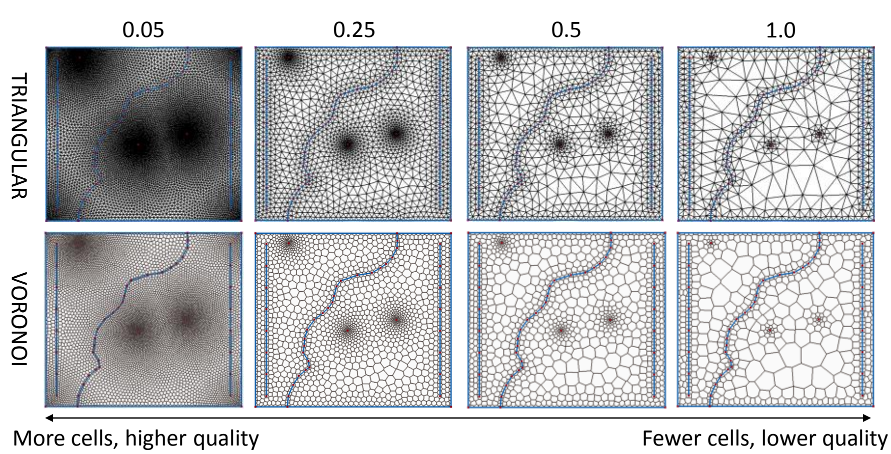
# Optimisation Algorithm

5. Reduce target sizing ratio by the reduction factor (to a min of 1.0)
6. Repeat steps 3-5 (refine and Lloyd relax) until all constrained edges and triangles have a sizing ratio ≤ 1.0
7. Perform a final Lloyd relaxation step (as per step 4) to a tighter movement threshold

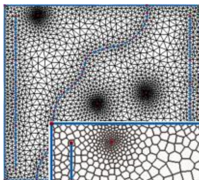




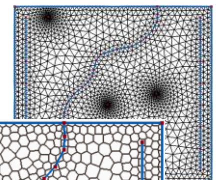

# Optimisation: Distance (Grading) Factor



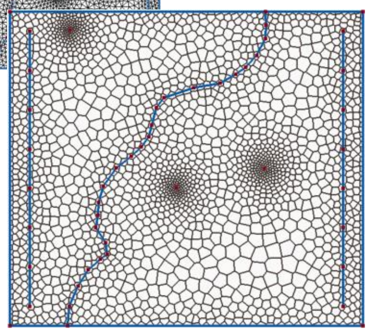
# Optimisation: Quality Preset

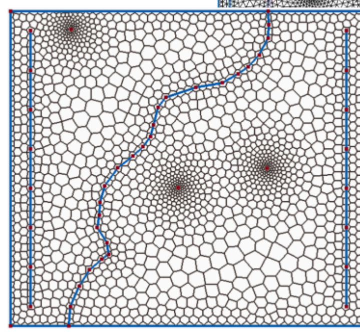


Low Quality (1)
4616 Triangles
3 Seconds



High Quality (6)
3728 Triangles
5 Minutes





Faster, more cells, lower quality

Slower, fewer cells, higher quality

# Tutorial 1: Mesh Generation

This tutorial explores a number of techniques for producing a high-quality Voronoi mesh. You will learn how to bring in a polygonal model extent boundary, streamlines and bores, and manipulate these using AlgoMesh's built-in polyline editing and resampling tools. You will also learn how to control cell sizing in different areas of the mesh, and explore the effects of different combinations of parameters to AlgoMesh's mesh generation algorithms.

Various sections of the tutorial instructions specify a "quick start" file. If you follow all the instructions through from the start, you can ignore these. However, if you get stuck in a particular section or want to jump ahead to learn about a different topic, you can skip the steps leading up to a section and load one of these AlgoMesh project files instead, which will start you off precisely where you need to be to begin the corresponding section.

### Input Data Files

The following files are included in the Tut1 folder for use with this tutorial. They are referred to at various stages throughout the step-by-step tutorial instructions.

| | |
|---|---|
| **bounding_box.xy** | A text file containing four corner points of a rectangle making up an outer bounding box of the model domain. |
| **Extent.mdd** | Polygon delineating the precise desired extent of the model. |
| **Rivers.shp** | Polyline shapefile containing streamlines to be included in the model mesh, extending to the model extent. |
| **Rivers_trim.shp** | Polyline shapefile containing streamlines to be included in the model mesh, trimmed away from the model extent. |
| **production_bores.shp** | Point shapefile containing the locations of pumping bores to be included in the model. |
| **observation_bores.shp** | Point shapefile containing observation locations of interest in the model. |
| **production_bore_polygons.shp** | Polygon shapefile containing circles buffered at a small radius around each of the production bores. These are used to constrain cell sizing in the vicinity of the bores. |
| **Alluvium.shp** | Polygon shapefile containing the outline of an alluvial region around the stream network in the model. This is used to constrain cell sizing within the alluvium. |

# Step-by-Step Instructions

### Bring in the Model Boundary

1. Start AlgoMesh.
2. Go to File->Import Mesh Geometry, choose bounding_box.xy. A rectangular extent of the model domain is shown. The bounding_box.xy file is simply a text file containing four pairs of X, Y coordinates representing the outermost bounds of the model domain.
3. Right-click Spline Sets, and choose Import GIS Polylines as Splines.
4. Find and select Extent.mdd – this contains a polygon of the actual boundary that will be used as the extents of the model.
5. Change the boundary to a linear polyline, instead of the default spline curve:
    a. Right-click the Extent spline set, and choose Set Properties for All Splines in Set.
    b. Set spline type to Linear.

    c. Click Apply to All in Set.
    d. Close the properties window.
6. Bake the boundary polyline into the mesh:
    a. Right-click the Extent spline set, and choose Add to Mesh Now and Deactivate Set -> Original Mesh and Generated Mesh.
7. In the Edit menu, ensure that the Edit Mesh mode is selected (ticked).
8. In the View menu, ensure that Triangles are visible (ticked).
9. Left-click on a triangle inside the bounding box but outside the model extents polygon.
10. Press the minus key '-' to remove the area containing this triangle from the model domain (the area of triangles in between the model extents polygon and the outer bounding box). The triangles in this area should turn light grey.
11. Press Escape to deselect the triangle.
12. Save your progress:
    a. Go to File->Save Project
    b. Choose an appropriate location and name for the project file, such as model_base.amproj and click Save.

### Import Streamlines and Wells (Quick Start: tut1-stage1.amproj)

13. Right-click Spline Sets, and choose Import GIS Polylines as Splines.
14. Select Rivers.shp.
15. Repeat to bring in production_bores.shp and observation_bores.shp.
16. Right-click the Rivers spline set and choose Set Properties for All Splines in Set.
17. Change spline type to Linear.
18. Choose a reasonable cell size for the river cells (say 50 m), and change edge length at start and edge length at end to that number (note: enter the number without the unit, e.g. 50, not 50m).
19. Click Apply to All in Set.
20. Close the properties window.

### Show the Voronoi Grid

21. In the View menu, turn Voronoi Cells on.
22. Also in the View menu, turn Triangles off.

### Try a Refinement Run and Clean the Streamline Geometry (Quick Start: tut1-stage2.amproj)

23. In the Refinement tab, with a minimum angle constraint of 30 degrees, click Refine Mesh. A mesh will be generated using the refinement algorithm.
24. Zoom the view in where one of the streamlines meets the southern boundary. Notice that there is significant over-refinement in this area (many small cells). This is because end of the streamline ends very close to an edge of the boundary polygon. Originally this point may have been positioned precisely on the edge, but rounding errors often make it impossible to match the edge exactly. There are three options for cleaning up the geometry in this scenario:
    a. Trim the stream polyline away from the boundary polygon in AlgoMesh. This can be done by activating Edit->Edit Splines mode, clicking on the last vertex in the streamline and pressing the Delete key.

   b. *(Recommended)* Trim the stream polylines away from the boundary polygon in GIS and recreate a new shapefile to bring back into AlgoMesh. There is a sample file called Rivers_trim.shp which you can use for this purpose. Import Rivers_trim.shp as a spline set and resample as a linear polyline at 50m intervals using the same process you used above, then remove the old Rivers spline set.
   c. Add a new vertex in the boundary polyline at precisely the position of the streamline endpoint (the intersection between the endpoint and the boundary edge). This would involve repeating the above steps to recreate the model boundary.
25. There are two spots on the southern boundary where the streamlines end too close to the boundary polygon; make sure both of these have been trimmed using one of the methods above.
26. Click Refine Mesh again. A new mesh should be generated that is no longer over-refined at the boundaries.
27. You may wish to turn off visibility of Splines at this stage (View->Splines) so that you can more clearly see the points that have been added to the mesh.

### Try an Optimisation Run (Quick Start: tut1-stage3.amproj)

28. Now that refinement is producing a reasonable-looking mesh, we can try an optimisation run, which will take a bit longer, but which tends to produce higher-quality cell geometry.
29. Go to the Optimisation tab on the right-hand side of the AlgoMesh window.
30. Turn on Global max. edge length and give it a value of 500.
31. Click Start Optimisation, leaving all the parameters at their default values.
32. Wait for the optimisation to complete. You should see a well-graded mesh with smaller elements around the streams and larger elements in the open areas of the model.
33. Experiment with other values of the distance (grading factor) and re-run the optimisation to see how this affects the mesh. Try a smaller value (e.g. 0.1) and a larger value (e.g. 1.0). If at any time you are not happy with how the mesh is looking during the optimisation process, you can stop it prematurely by clicking Stop Optimisation.
34. With a moderate value of distance (grading) factor (say 0.4), experiment with different quality presets and note the effect on the resulting mesh. Note in particular that the number of triangles (and ultimately the number of cells) is much smaller on higher quality settings, but the mesh takes longer to generate.

### Add Refinement for Production Bores (Quick Start: tut1-stage4.amproj)

35. One way to refine the mesh around bores is to turn on the option "Max. edge length at constrained points" and specify a reasonable cell size value for it, e.g. 10. Try this and run the optimisation again. If mesh generation is taking too long, stop it, set a lower quality preset and run it again. You will see that the resulting mesh is refined around every single-point feature in the model.
36. If you don't want to refine around every point – just some of them – then you need to use a more specific method of cell size mapping. Here, we would like to refine around production bores but we don't need the refinement around observation bores. We will use a polygon-based Mesh Cell Edge Length mapping for this. First turn "Max. edge length at constrained points" off.
37. Right-click Property Polygon Sets and select Import GIS Polygons.
38. Find and load the file production_bore_polygons.shp. This file contains a set of small buffered polygons – each one containing a production bore location.
39. Right-click the new production_bore_polygons set, and choose New Property.

40. Give the new property the name CellSize, and a default value of 10. Each of the polygons in the set now have a field called CellSize with the default value of 10 associated with them.
41. Map the new field CellSize to Mesh Cell Edge Length:
    a. Right-click the CellSize field.
    b. Select Map to Mesh Variable as Overlay->Mesh Cell Edge Length.
42. We now need to tell the Edge Length Field Generation process to take this mapping into account. First, find the latest EdgeLengthField property TIN that has been generated – it will have an EdgeLength entry underneath it that is mapped to the Mesh Cell Edge Length property.
43. Delete this EdgeLengthField property TIN. We do this so that the next time AlgoMesh generates an edge length field for the optimisation, it does not re-use any of the old cell size information. Alternatively, if you have no other property TINs loaded, you can remove all generated edge length fields by right-clicking the Property TINs header and choosing Remove All Property TINs.
44. Turn on both edge length mapping options in the optimisation tab: "Use existing edge length mapping as max. at boundaries" and "Use existing edge length mapping as max. elsewhere".
45. Click Start Optimisation to re-run the optimisation using the new edge length constraints, and wait until it completes. There should now be refinement down to 10m cells around the production bores, but observation bores are left at the normal mesh resolution.

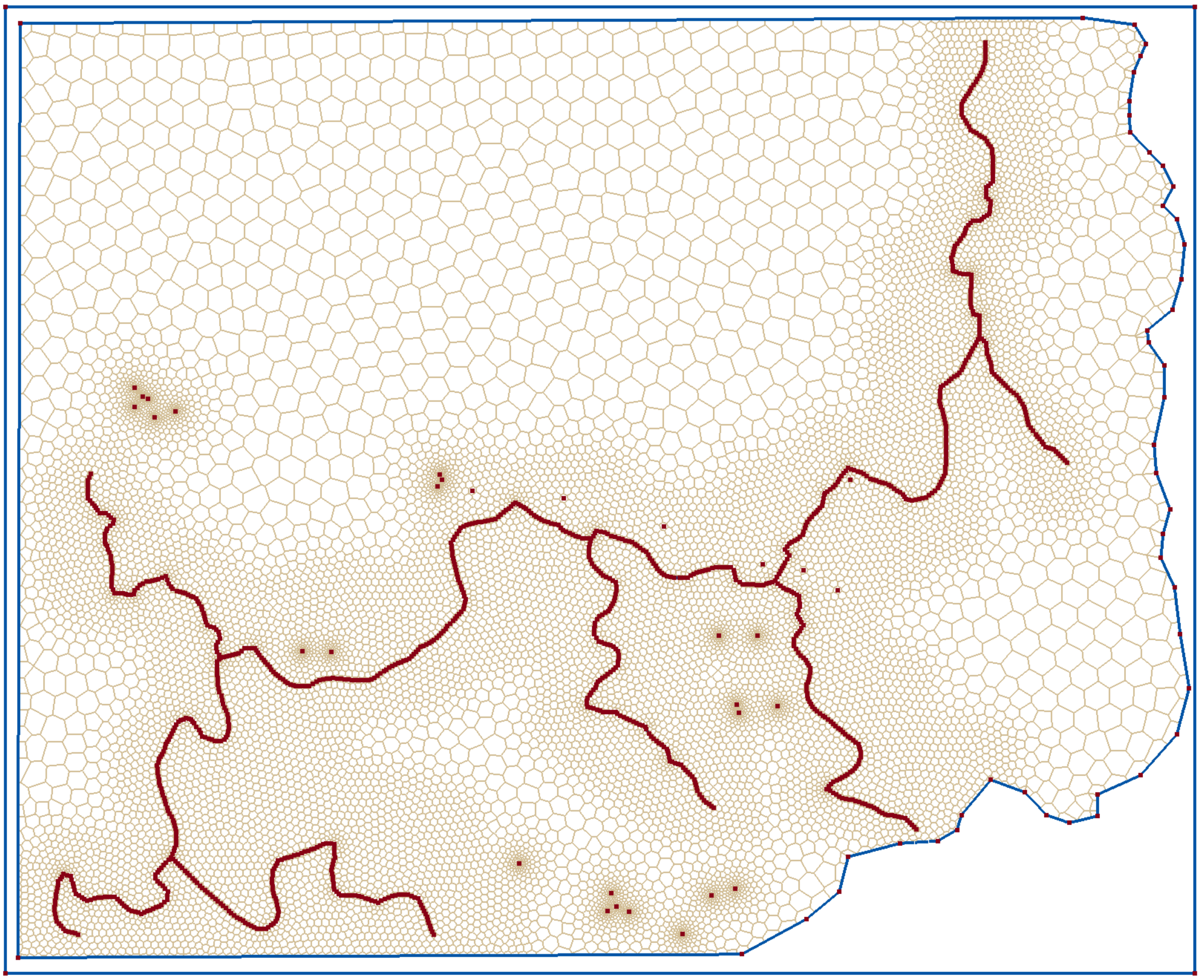
### Add Refinement in Alluvium (Quick Start: tut1-stage5.amproj)

46. You may also have larger regions of your model where you would like to restrict cell sizes to improve simulation accuracy. In this case we are going to bring in a polygon delineating approximate outer extents of an alluvial region, and tell AlgoMesh to refine to smaller cells within this region. First, right-click Property Polygon Sets and choose Import GIS Polygons.
47. Load Alluvium.shp. Alluvium is added as a new set below the production_bore_polygons set.
48. As before, add a new property to the Alluvium set called CellSize, and give it a default value of 150.
49. Map the Alluvium set's CellSize property to Mesh Cell Edge Length.
50. Remove the latest EdgeLengthField property TIN.
51. Turn on both "Use existing edge length mapping" options once again, and ensure "Auto-generate at start of optimisation" is also turned on.
52. Click "Generate Edge Length Property TIN". This will generate the new edge length field, taking into account the new cell sizing constraint within the alluvium.
53. Once the edge length field has been generated, and before starting the optimisation, remove the Alluvium mapping to cell size. This will ensure that the 150m cell size mapping does not override smaller cell size values that were produced by the edge length field generation algorithm; if we leave the mapping active during optimisation, the polygon mapping will take precedence over the edge length field.
    a. Under the Alluvium polygon set, right-click the Mesh Cell Edge Length entry below CellSize.
    b. Choose Remove Mapping.

### Generate the Final Mesh (Quick Start: tut1-stage6.amproj)

54. In the Optimisation tab, ensure "Auto-generate at start of optimisation" is turned off. This ensures that the existing edge length field that is mapped to Mesh Cell Edge Length will be used.

55. Click Start Optimisation to generate the mesh. If you notice that the optimisation appears to become stuck in a loop and doesn't progress, stop it, turn on "Max. Lloyd iterations (refine)" with a value of say 50, and then start the optimisation again. This forces the algorithm to progress to the next refinement stage after 50 smoothing iterations have completed.
56. Once the optimisation has completed (or if you choose to stop it early), turn off Edit->Mesh Generation Restarts from Original Mesh Domain).
57. Now go to the Refinement tab, and set a modest minimum angle constraint – say 26 degrees.
58. Click Refine Mesh. This step adds a small number of additional cells to the model, but is important to fix up any concave or poorly-shaped cells that are left over after the optimisation process completes.
59. You now have a complete 2D mesh. Save your progress by going to File->Save Project, choosing an appropriate new name for the project file, e.g. model-mesh.amproj and clicking Save.
60. You may optionally wish to export a shapefile of the mesh for viewing and manipulation in GIS software. Do this by going to File->Voronoi Mesh->Save Cell Geometry. Type a filename, such as meshv1.shp and click Save. AlgoMesh will create a polygon shapefile containing all of the Voronoi cells in the model, and a corresponding data table with the cells' indices and cell centre positions.



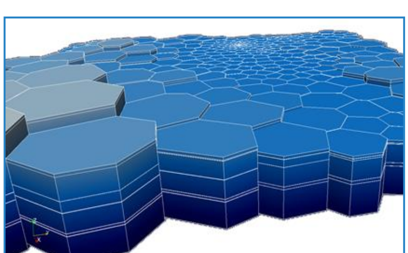
# Building Multi-layered MODFLOW-USG Models

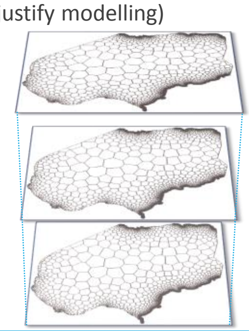
SESSION 2: MODFLOW-USG MODEL BUILDING
(STEADY-STATE, PART 1)

HYDRO ALGORITHMICS

# USG Layering Overview

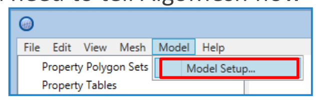
- USG uses **prismatic cells**, just like regular MODFLOW
- AlgoMesh always uses the same 2D mesh structure for every layer *(an AlgoMesh restriction, not a USG restriction)*
  - BUT some **cells may be removed completely** from the model outside the geological layer extents (or where too thin to justify modelling)
  - More on this later in the session…

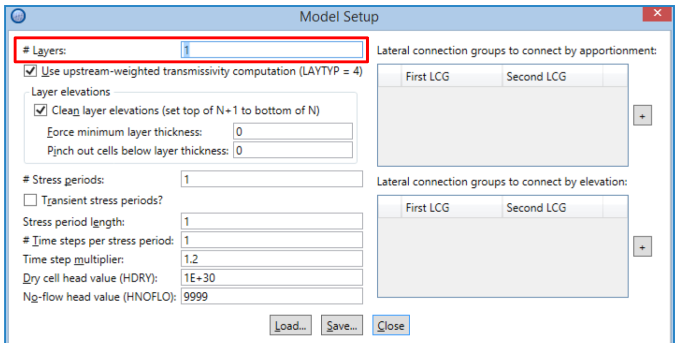




# Building a 3D Model: Model Setup

- Once you have a satisfactory 2D mesh, you need to tell AlgoMesh how many layers you need in your 3D model
- Optional elevation cleanup



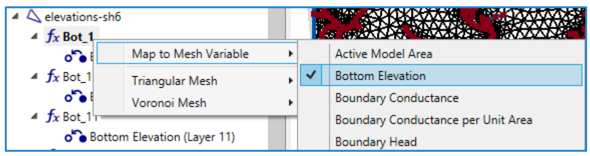



# Building a 3D Model: Assigning Elevations

- USG needs top and bottom elevations for each cell (as per regular MF)
- In AlgoMesh, just assign top of layer 1 and bottom of layers 1 to N
  - AlgoMesh will apply bottom of layer 1 as top of layer 2, and so on
- Two mechanisms for assigning elevations
  - Property TINs (triangulated grid of sample points)
  - Property tables (CSV file specifying exact elevation values for each cell)

# Building a 3D Model: Assigning Elevations by TIN

- To assign elevations by property TIN:
  1. In GIS software, **load a raster DEM** for each layer
  2. **Export X, Y, Z values** from each raster to a comma-separated XYZ file
  3. Create a **new property TIN** in AlgoMesh from each XYZ file
  4. **Map the field** in each property TIN to the corresponding layer top or bottom elevation




- Property TINs with multiple variables may be built by creating a Mesh Domain Description (.MDD) file – manually or by a scripted process

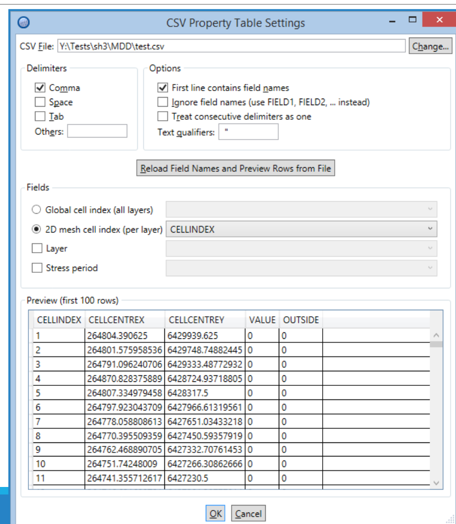


# Building a 3D Model: Assigning Elevations by CSV

- To assign elevations by property table (CSV):
  1. Export mesh cell geometry from AlgoMesh to GIS (as polygon shapefile)
  2. In GIS, create and populate fields for elevations at each cell
     - Useful for applying a more rigorous method to choose elevations, e.g. areal averaging over cell polygon on raster DEM
  3. Export data table containing elevations from GIS as CSV
     - Each record should contain Mesh Cell Index, Layer 1 Top, Layer 1 Bottom, Layer 2 Bottom, etc.
  4. Import CSV into AlgoMesh as a property table
  5. Map each field to corresponding layer top/bottom elevation

HYDRO ALGORITHMICS

# Building a 3D Model: Assigning Elevations by CSV

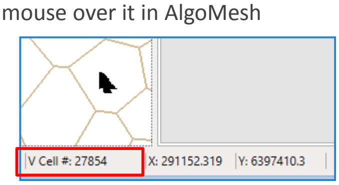





HYDRO ALGORITHMICS

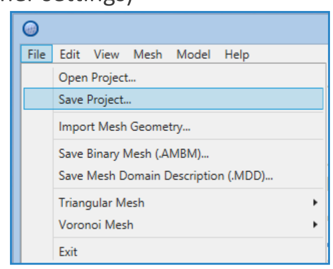
# Cell Numbering

- Unstructured grids have **no "natural" ordering**, unlike regular MODFLOW with rows and columns
  - Not obvious which cell is assigned what cell index; need to inspect tables
- Export mesh cell geometry to shapefile to AlgoMesh's cell numbering for a single layer (called the **2D mesh cell index**)
  - Includes data table with cell index => cell centre X, Y coordinates
  - OR inspect a single cell's index by hovering the mouse over it in AlgoMesh
- With no removed cells, cell numbering is sequential through layers 1 to N
  - e.g. Layer 1 cells are numbered from 1 to (# cells per layer)
    Layer 2 cells are numbered from (# cells per layer) + 1 to 2 * (# cells per layer)



# Cell Numbering

- **Be aware:**
  - Cell numbering may change whenever you make a change to the mesh (i.e. change geometry or start refinement or optimisation methods)
- **Save as .amproj** to retain numbering (and other settings)
- If you change the mesh, you will need to **rebuild any CSV property table data** to correspond to the new numbering of cells



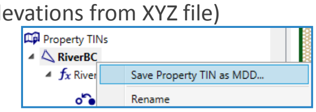



# AlgoMesh File Types

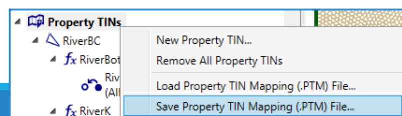
- AlgoMesh project file (.amproj)
  - Stores almost everything:
    - Original and generated meshes
    - Model setup options
    - Edit and View menu options
    - Property polygon sets
    - Splines
    - Property TINs
    - Property table settings
    - All mesh variable mappings
  - Exceptions:
    - Does not store the actual CSV files associated with a property table (only the path to the file)
      - May need to adjust CSV paths when copying .amproj file to another machine
    - Does not store Python scripts (will change in a future version)

# AlgoMesh File Types

- Mesh domain description file (.mdd)
  - Stores vertices and edges of a single mesh or property TIN
  - Note that the mesh is retriangulated when loaded from .mdd, which may change cell numbering (use .amproj instead for model meshes)
  - Optionally stores vertex properties (e.g. elevations from XYZ file)
  - Optionally stores splines



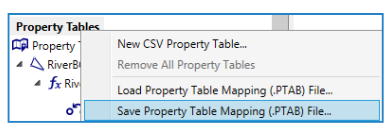

- AlgoMesh binary mesh file (.ambm)
  - Similar to .mdd, but also stores the triangles, so no retriangulation needed
- Property TIN mapping file (.ptm)
  - Stores file paths of a collection of property TINs (as .mdd files)
  - Stores mesh variable mappings to property TIN properties



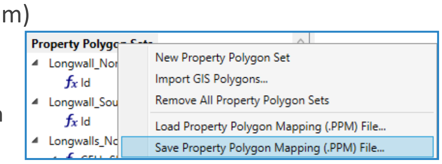



# AlgoMesh File Types

- Property table mapping file (.ptab)
  - Similar to .ptm file, but stores paths to CSV files
  - Stores settings for each CSV (delimiters, cell index and layer fields, etc.)
  - Stores mesh variable mappings to property table properties

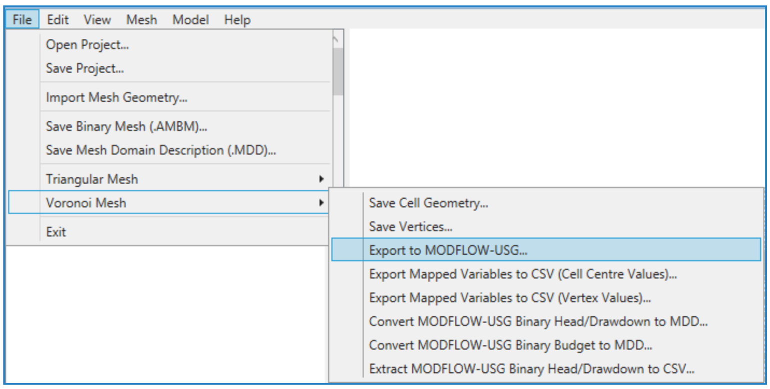



- Property polygon mapping file (.ppm)
  - Stores polygon sets
  - Stores geometry of polygons
  - Stores property data associated with each polygon
  - Stores mesh variable mappings to polygon properties



# MODFLOW-USG Discretisation

- Exporting from AlgoMesh to MODFLOW-USG produces an unstructured discretisation file (DISU)





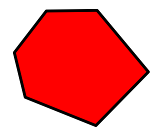

# MODFLOW-USG Discretisation

- The DISU file specifies all the geometric information USG needs
  - **NLAY** ◦ Number of layers
  - **NODELAY** ◦ Cell count per layer (may vary)
  - **Top,Bot** ◦ Cell top and bottom elevations
  - **Area** ◦ 2D cell area
  - **IAC,JA** ◦ Connections between cells
  - **IVC** ◦ Vertical/horizontal connection types
  - **CL12** ◦ Perpendicular distances to interfaces
  - **FAHL** ◦ Interface areas

Hydro Algorithmics

# MODFLOW-USG Discretisation

- The DISU file specifies all the geometric information USG needs
  - **NLAY** Number of layers
  - **NODELAY** Cell count per layer (may vary)
  - **Top,Bot** Cell top and bottom elevations
  - **Area** 2D cell area
  - **IAC,JA** Connections between cells
  - **IVC** Vertical/horizontal connection types
  - **CL12** Perpendicular distances to interfaces
  - **FAHL** Interface areas

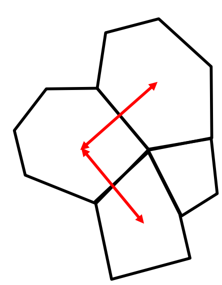




Hydro Algorithmics

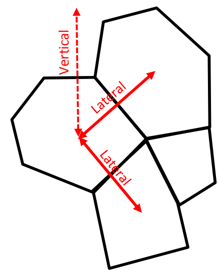
# MODFLOW-USG Discretisation

- The DISU file specifies all the geometric information USG needs
  - **NLAY** Number of layers
  - **NODELAY** Cell count per layer (may vary)
  - **Top,Bot** Cell top and bottom elevations
  - **Area** 2D cell area
  - **IAC,JA** Connections between cells
  - **IVC** Vertical / lateral connection types
  - **CL12** Perpendicular distances to interfaces
  - **FAHL** Interface areas



# MODFLOW-USG Discretisation

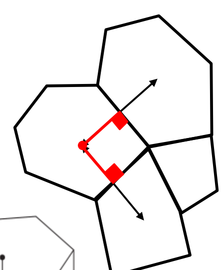
- The DISU file specifies all the geometric information USG needs
  - **NLAY** ◦ Number of layers
  - **NODELAY** ◦ Cell count per layer (may vary)
  - **Top,Bot** ◦ Cell top and bottom elevations
  - **Area** ◦ 2D cell area
  - **IAC,JA** ◦ Connections between cells
  - **IVC** ◦ Vertical / lateral connection types
  - **CL12** ◦ Perpendicular distances to interfaces
  - **FAHL** ◦ Interface areas

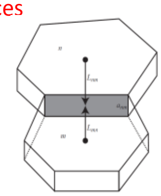





# MODFLOW-USG Discretisation

- The DISU file specifies all the geometric information USG needs
  - **NLAY** ◦ Number of layers
  - **NODELAY** ◦ Cell count per layer (may vary)
  - **Top,Bot** ◦ Cell top and bottom elevations
  - **Area** ◦ 2D cell area
  - **IAC,JA** ◦ Connections between cells
  - **IVC** ◦ Vertical / lateral connection types
  - **CL12** ◦ Perpendicular distances to interfaces
  - **FAHL** ◦ Interface areas

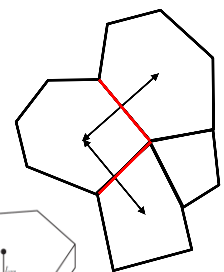


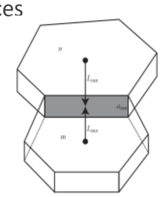


*(Panday et al., 2012)*

# MODFLOW-USG Discretisation

- The DISU file specifies all the geometric information USG needs
  - **NLAY** Number of layers
  - **NODELAY** Cell count per layer (may vary)
  - **Top,Bot** Cell top and bottom elevations
  - **Area** 2D cell area
  - **IAC,JA** Connections between cells
  - **IVC** Vertical / lateral connection types
  - **CL12** Perpendicular distances to interfaces
  - **FAHL** Interface areas
    = (Shared edge length) *
    (Average layer thickness)





*(Panday et al., 2012)*



# MODFLOW-USG Discretisation

- The DISU file does NOT specify:
  - Cell perimeter coordinates
  - Cell centre coordinates
- Use the *de facto* standard **grid specification file (GSF)** for these
  - AlgoMesh always outputs a GSF alongside the DISU
  - Allows you to use AlgoMesh outputs with the PEST groundwater utilities and (later) Groundwater Vistas
  - Documentation available with PEST groundwater utilities

Hydro Algorithmics

# MODFLOW-USG Discretisation

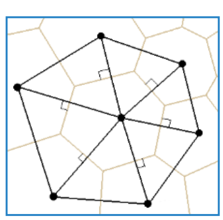
- The DISU file also specifies the temporal discretisation
  - Number of stress periods
  - Type of each stress period (steady-state or transient)
  - Length of each stress period
  - Number of time steps and time step length multiplier for each stress period
- AlgoMesh currently outputs the same type, length and number of time steps for all stress periods
  - Need to edit DISU file and change these manually

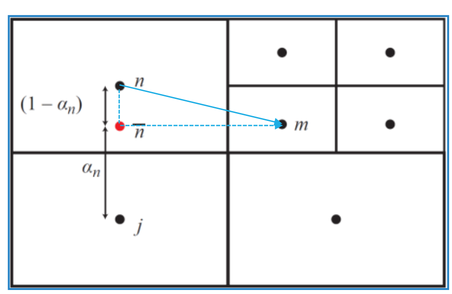


Hydro Algorithmics

# Ghost Node Correction

- Control-Volume Finite Difference (CVFD) formulation assumes:
  - Cells are convex
  - The flow line between each pair of adjacent cells must:
    1. Be perpendicular to the shared edge; and
    2. Bisect the shared edge
  - Voronoi grids give #1 for free
  - Can use **ghost nodes** for #2 (GNC)






*(Panday et al., 2012)*

# Ghost Node Correction

- AlgoMesh produces a GNC file using inverse-distance weighting (IDW) to obtain an interpolated head from neighbouring nodes in the underlying triangulation
  - Not proven
  - Not recommended!
- General guidance:
  - GNC is probably not important for Voronoi grids in most cases
  - GNC tends to make it harder to converge on a solution
  - Only important in presence of a severe head gradient (laterally). In this case:
    - Refine to **smaller grid cells** in problem area (reduces error in head approximation); and/or
    - Use a **structured sub-grid** with perfect hexagonal cells (guarantees correct head is used)

# Cutting Out Unnecessary Model Cells

SESSION 2: MODFLOW-USG MODEL BUILDING (STEADY-STATE, PART 1)

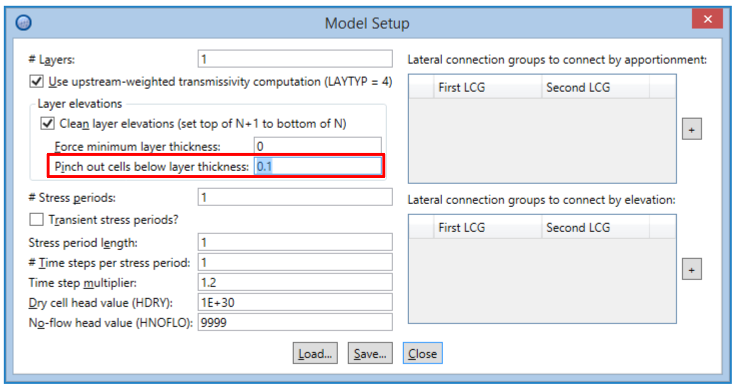
HYDRO ALGORITHMICS

# Removal of Cells

- One of USG's major benefits is the ability to remove unneeded cells
  - Only need cells to extents of each layer, not whole model
  - Can model geological outcrops, subcrops, pinchouts
- Three mechanisms (can be combined):
  1. Tell AlgoMesh to **remove all cells with less than a given thickness**
  2. Delineate **polygons of active and inactive cells**, and map in AlgoMesh to 1 and 0 values (resp.) of the "Active Model Area" mesh variable
  3. Use your own **manual or scripted process** outside of AlgoMesh to determine which cells are active and inactive, then bring this data into AlgoMesh as a **CSV property table**

# Cell Removal: By Thickness

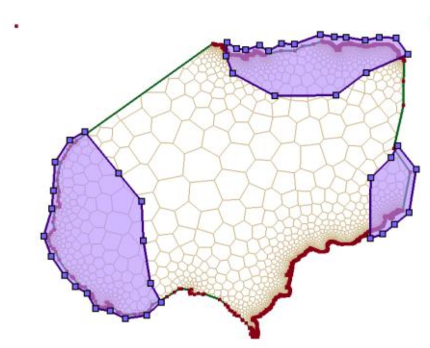
- The **Model Setup** window allows automatic pinchout of thin cells
  - Set top elevation == bottom elevation in areas where layer does not exist





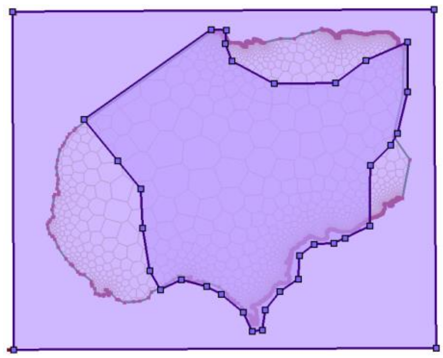

# Cell Removal: By Polygon

- The **Active Model Area** variable determines which cells are included
  - Active Model Area == 1 at cell centre → cell included in model
  - Active Model Area == 0 at cell centre → cell removed from model
- Option A:
  - **Specify only inactive areas** of each layer (map Active Model Area to 0 inside polygons)
- Option B:
  - Map Active Model Area to 0 for entire model and **specify only active area** (map Active Model Area to 1 inside polygon)
  - Ensure active area polygon set is **above** entire model polygon set in Property Polygon Sets list



## Cell Removal: By Polygon

- The **Active Model Area** variable determines which cells are included
  - Active Model Area == 1 at cell centre → cell included in model
  - Active Model Area == 0 at cell centre → cell removed from model
- Option A:
  - **Specify only inactive areas** of each layer (map Active Model Area to 0 inside polygons)
- Option B:
  - Map Active Model Area to 0 for entire model and **specify only active area** (map Active Model Area to 1 inside polygon)
  - Ensure active area polygon set is **above** entire model polygon set in Property Polygon Sets list

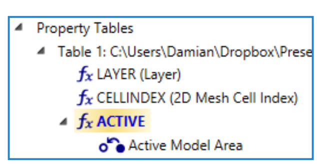




## Cell Removal: Custom Mapping

- As with elevation mapping, you can **specify on a cell-by-cell basis** which cells should be included and which should be removed
  - Create CSV file with (Layer, 2D Mesh Cell Index, Active) records
  - Can limit to just those cells that are inactive (Active == 0), as default value for Active Model Area is 1

```
Layer,CellIndex,Active
1,52,0
1,53,0
1,133,0
1,134,1
1,135,1
1,136,1
1,137,0
1,138,0
1,139,0
1,140,0
```



# Cell Numbering

- Inactive cells are completely removed from the USG model
  - This affects cell numbering!
- Regardless of the mechanism used, AlgoMesh outputs two files:
  - **<MODEL PREFIX>-layer-to-global-index.csv**
    *(Layer + 2D Mesh Cell Index → Global Cell Index)*
  - **<MODEL PREFIX>-global-to-layer-index.csv**
    *(Global Cell Index → Layer + 2D Mesh Cell Index)*



# Layer to Global Cell Index

- **<MODEL PREFIX>-layer-to-global-index.csv**
  *(Layer + 2D Mesh Cell Index → Global Cell Index)*

```
Layer,LayerCellIndex,GlobalCellIndex
1,1,1
1,2,2
1,3,3
1,4,4
1,5,5
1,6,6
1,7,7
1,8,8
1,9,9
1,10,10
1,11,11
1,12,12
1,13,13
1,14,14
1,15,15
1,16,16
1,17,17
1,18,18
```

```
2,344,36330
2,345,36331
2,346,36332
2,347,-1
2,348,-1
2,349,-1
2,350,-1
2,351,-1
2,352,-1
2,353,-1
2,354,-1
2,355,-1
2,356,-1
2,357,-1
2,358,36333
2,359,36334
2,360,36335
2,361,36336
```

```
15,35977,495656
15,35978,495657
15,35979,495658
15,35980,495659
15,35981,495660
15,35982,495661
15,35983,495662
15,35984,495663
15,35985,495664
15,35986,495665
15,35987,495666
```

# Global to Layer Cell Index

- **<MODEL PREFIX>-global-to-layer-index.csv**
  *(Global Cell Index → Layer + 2D Mesh Cell Index)*

```
GlobalCellIndex,Layer,LayerCellIndex
1,1,1
2,1,2
3,1,3
4,1,4
5,1,5
6,1,6
7,1,7
8,1,8
9,1,9
10,1,10
11,1,11
12,1,12
13,1,13
14,1,14
15,1,15
16,1,16
17,1,17
18,1,18
```

```
36330,2,344
36331,2,345
36332,2,346
36333,2,358
36334,2,359
36335,2,360
36336,2,361
36337,2,362
36338,2,363
36339,2,364
36340,2,365
36341,2,366
36342,2,367
36343,2,368
36344,2,369
36345,2,370
36346,2,371
36347,2,372
```

```
495656,15,35977
495657,15,35978
495658,15,35979
495659,15,35980
495660,15,35981
495661,15,35982
495662,15,35983
495663,15,35984
495664,15,35985
495665,15,35986
495666,15,35987
```

# Assigning Properties and Boundary Conditions

SESSION 3: MODFLOW-USG MODEL BUILDING (STEADY-STATE, PART 2)

## Steady-state Model Properties

- A steady-state model must have the following properties mapped for all layers, in addition to elevations:
  - Kx *(Note: AlgoMesh does not currently support anisotropy, so Ky == Kx)*
  - Kz
  - Initial Head


- Any cells with missing values will take a 0 value
  - Be careful to map entire model domain for all layers
- All three mapping mechanisms are supported for these:
  - Property polygon sets
  - Property tables
  - Property TINs

HYDROALGORITHMICS

# Boundary Conditions

- Several boundary condition (BC) packages are supported

| Package | Property |
|---|---|
| | Active Model Area |
| | Bottom Elevation |
| GHB | Boundary Conductance |
| | Boundary Conductance per Unit Area |
| | Boundary Head |
| CHD | Constant Head |
| DRN | Drain Conductance |
| | Drain Conductance per Unit Area |
| | Drain Elevation |
| EVT | ET Extinction Depth |
| | ET Maximum Flux |
| | ET Surface Elevation |

| Package | Property |
|---|---|
| RCH | Recharge |
| RIV | River Bottom Elevation |
| | River Conductance |
| | River Conductance per Unit Area |
| | River Stage |
| | Specific Storage |
| | Specific Yield |
| | Top Elevation |
| TVM | TVM Kx (End of Period) |
| | TVM Kz (End of Period) |
| | TVM Specific Storage (End of Period) |
| | TVM Specific Yield (End of Period) |
| WEL | Well Pumping Rate |



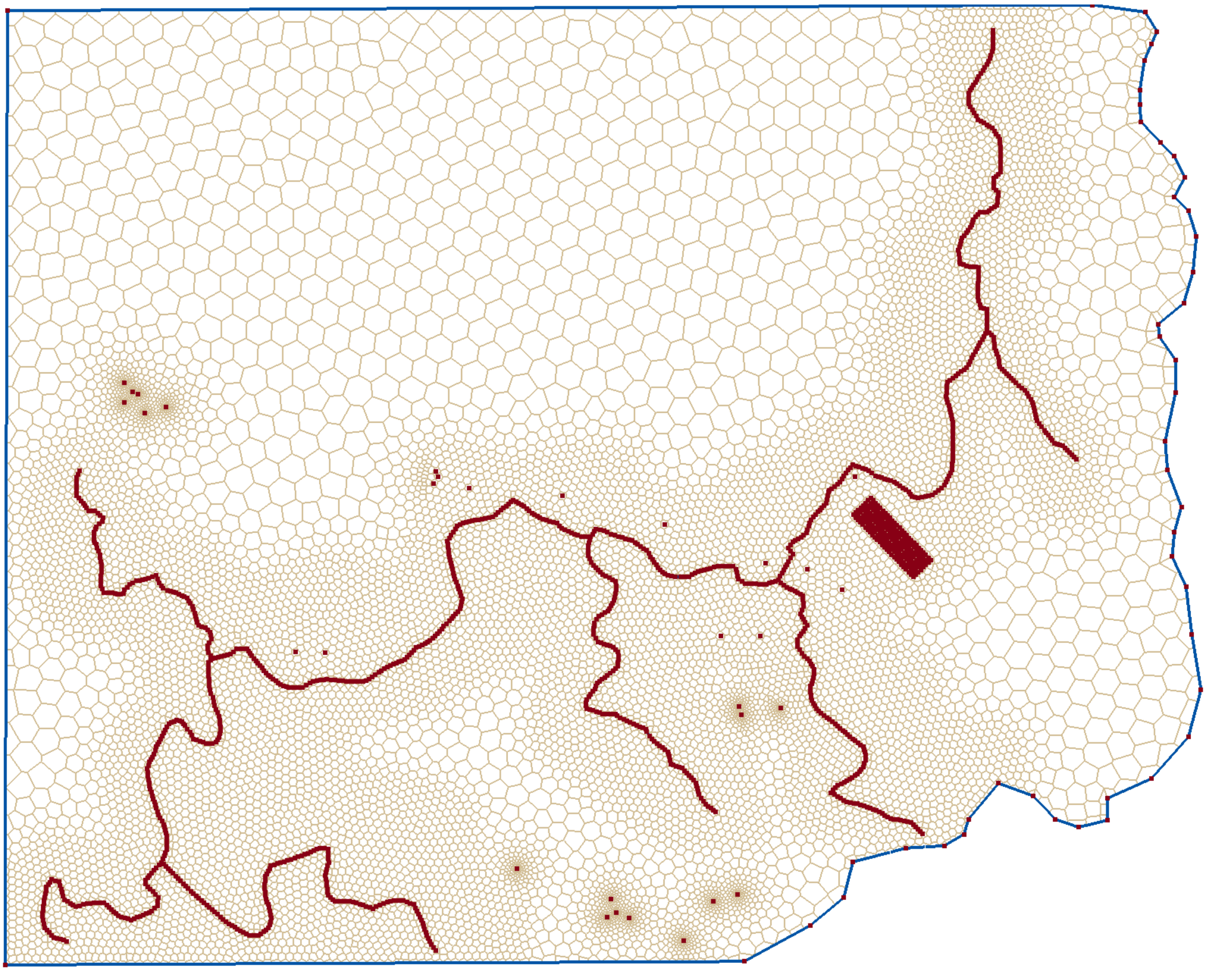
HYDROALGORITHMICS

# Boundary Conditions

- Same three mapping mechanisms supported
  - Property polygon sets
  - Property tables
  - Property TINs
- Any cells without a BC mapping have no BC applied
  - Exceptions: RCH and EVT
    (these always apply to entire top of model when active)
- Any cells with a BC mapping have the BC applied, even if the value is 0
  - Note the difference between mapping a **0 value** and **not mapping** any value (e.g. inside or outside polygons, inside or outside TIN, included or excluded from CSV file)

## Tutorial 2: Steady-State Model Building

In this tutorial, starting from a given mesh, you will build and run a complete steady-state MODFLOW-USG model. The mesh is similar to the one built in Tutorial 1, but with the addition of an underground longwall mine feature in the eastern part of the model domain.



### Model Layers

The model will contain seven layers, as detailed below.

| Layer | Description | Indicative Thickness | Kx | Kz | Ss | Sy |
|---|---|---|---|---|---|---|
| **1** | Regolith / Alluvium | 50m | 5 / 10 | 0.5 / 1 | 1e-5 / 1e-5 | 0.1 / 0.2 |
| **2** | Aquifer | 100m | 0.1 | 0.01 | 1e-5 | 0.1 |
| **3** | Aquitard | 25m | 1e-5 | 1e-6 | 1e-5 | 0.005 |
| **4** | Aquifer | 150m | 1e-4 | 1e-5 | 1e-5 | 0.02 |
| **5** | Coal Seam | 5m | 1e-2 | 1e-3 | 1e-6 | 0.01 |
| **6** | Aquifer | 25m | 1e-4 | 1e-5 | 1e-6 | 0.02 |
| **7** | Aquitard | 100m | 1e-5 | 1e-6 | 1e-6 | 0.005 |

### Boundary Conditions

WEL: A constant pumping rate of 1000 $m^3$/day is to be applied at all production bores.

RIV: A river boundary condition is to be applied to all cells along streamlines, with a constant bed conductivity of 0.01 m/day, a bottom elevation 10m below surface and a constant depth of 5m.

CHD: Constant head equal to the surface elevation is applied around the boundaries of the model.

RCH: Constant recharge of 4 x $10^{-4}$ m/day is applied across the whole domain.

EVT: Evapotranspiration at a constant maximum rate of 3 x $10^{-4}$ m/day and constant depth of 2m from surface elevation is applied.

### Input Data Files

The following files are needed to set up material property and boundary condition mappings.

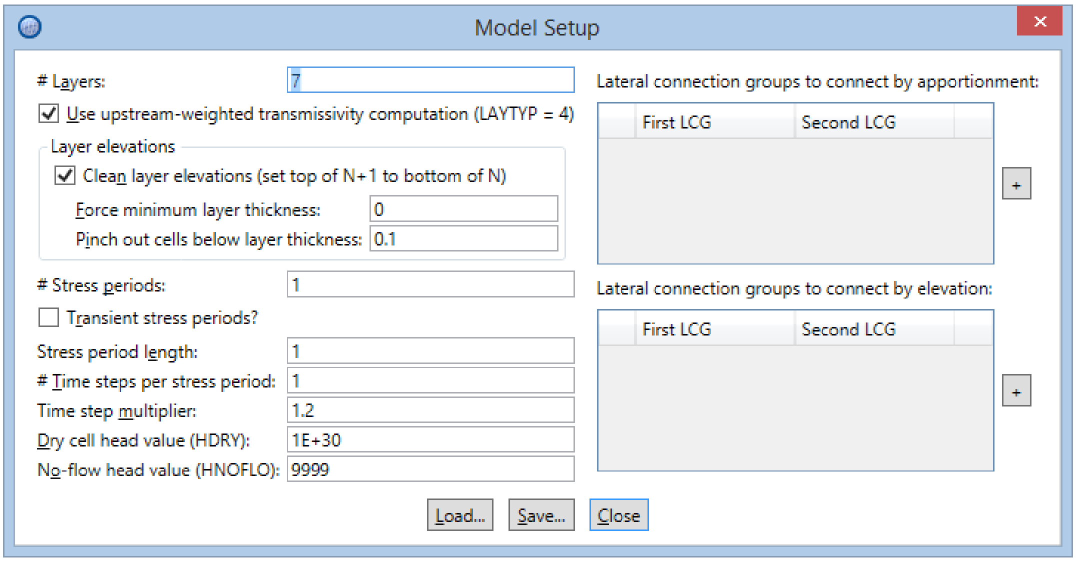
| | |
|---|---|
| **tut2-start.amproj** | AlgoMesh project file containing the starting point for this tutorial: a pre-generated Voronoi mesh with refinement in the alluvium and at production bores, and an induced structured grid in the location of a proposed underground longwall mine (to be used in a subsequent tutorial on transient modelling). |
| **elevations.shp** | Point shapefile containing top elevation of layer 1 and bottom elevations of layers 1 to 7. This should be loaded as a property TIN for elevation mapping. |
| **river.mdd** | Property TIN restricted to river cells only, giving river bottom elevations, river K (conductance per unit area) and river stage. |
| **CHD_all_poly.shp** | Polygon shapefile containing constant head values to be mapped around the outer boundaries of the model. |
| **Alluvium.shp** *(included in initial .amproj)* | Polygon shapefile containing an outline of the alluvial region for layer 1 of the model. Contains Kx and Kz properties to be mapped for this region (as well as Ss and Sy properties to be used later in the transient model). |
| **Entire_model_area.shp** *(optional; alternatively you may create this manually within AlgoMesh)* | Polygon shapefile covering the entire model area. Contains pre-populated Kx, Kz, Ss, Sy material properties for layers 1 to 7 (excluding the alluvium), and values for recharge, ET rate and ET depth. |
| **production_bore_polygons.shp** *(included in initial .amproj)* | Polygon shapefile covering production bore cells. Contains a WellQ property to be mapped to well pumping rate. |

## Step-by-Step Instructions

### Model Setup

1. Start AlgoMesh.
2. Go to File->Open Project, select tut2-start.amproj from the Tut2 folder and click Open.
3. Once the model has opened and the mesh is displayed, go to Model->Model Setup.
4. Set # Layers to 7
5. Ensure Clean layer elevations is turned on.

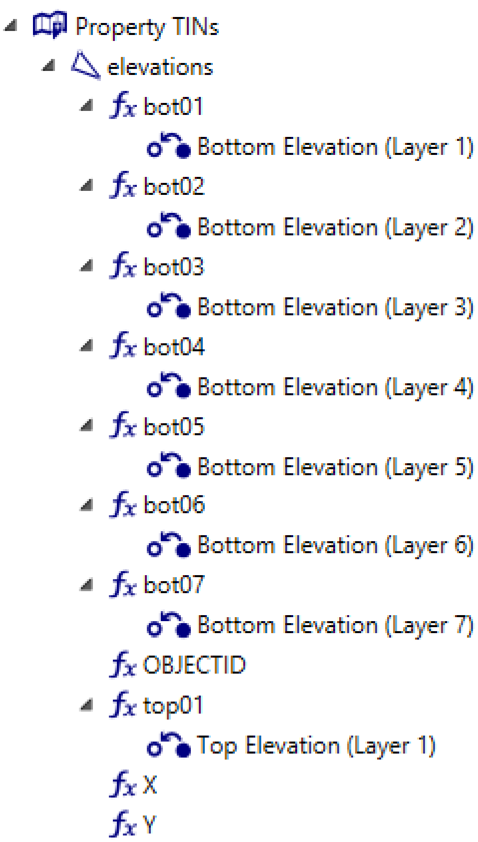
6. Set Pinch out cells below layer thickness to 0.1. Your Model Setup window should look like the one below.




7. Close the Model Setup window.

## Map Layer Elevations (Quick Start: tut2-stage1.amproj)

8. Right-click on Property TINs, and choose New Property TIN.
9. Select elevations.shp and click Open.
10. It may take a minute to load the point shapefile. For future usage, you may wish to save this file as an .MDD, which will load more quickly:
    a. Right-click the elevations layer and choose Save Property TIN as MDD.
    b. Give the MDD a name, such as elevations.mdd, and click Save.
11. Notice that the elevations property TIN has properties top01 and bot01 through bot07. Map these to the corresponding layer elevations:
    a. Right-click the top01 property, choose Map to Mesh Variable->Top Elevation.
    b. Ensure the mapping that appears is set to Layer 1. If not, right-click the mapping and choose Edit layers, type 1 and then hit Enter.
    c. Right-click the bot01 property, choose Map to Mesh Variable->Bottom Elevation.
    d. Ensure the mapping that appears is set to Layer 1.
    e. Right-click the bot02 property, choose Map to Mesh Variable->Bottom Elevation.
    f. Ensure the mapping that appears is set to Layer 2.
    g. Repeat steps e-f for bot03 through bot07. As there should be no other property TIN mappings to these elevations yet, if you do these in layer order, AlgoMesh will automatically increment the layer number each time you create a new mapping.

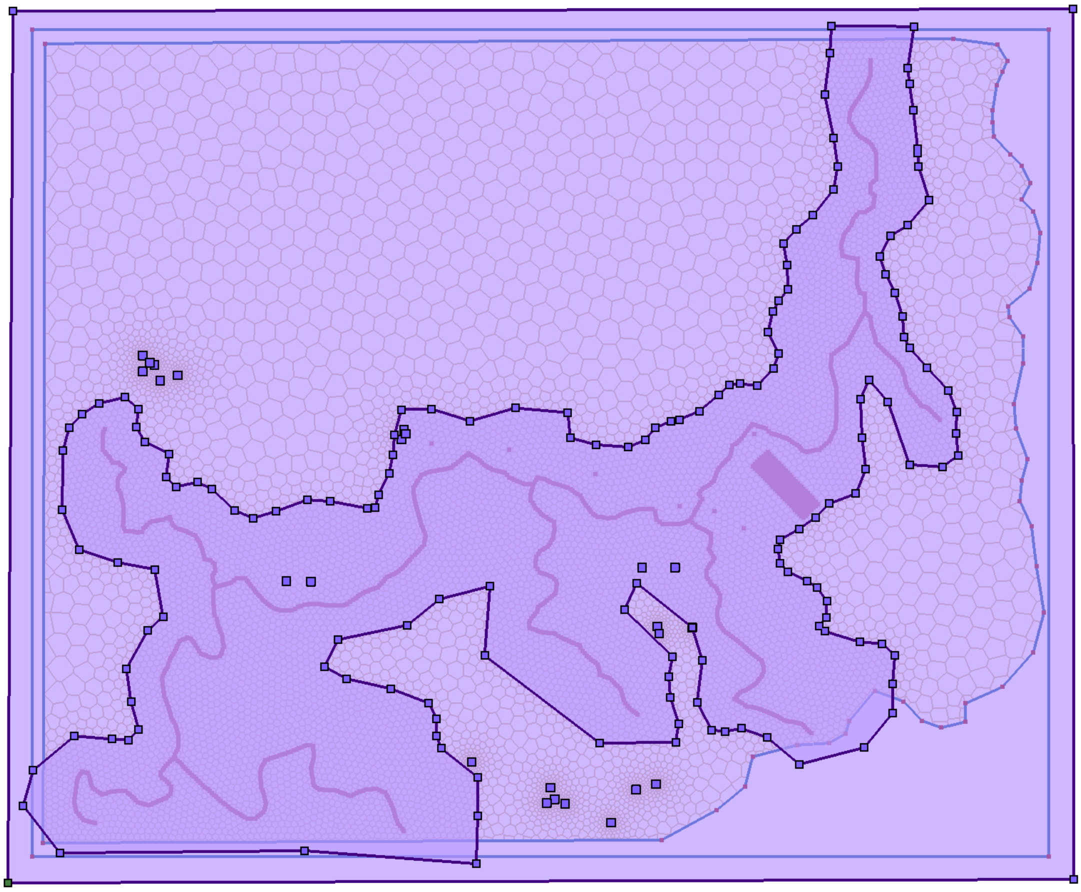



### Create Constant Property Value Polygon (Quick Start: tut2-stage2.amproj)

Follow these steps to manually create a polygon covering the whole model with properties for Kx and Kz for each layer.

*Alternatively, to save time, you may load the pre-populated shape file Entire_model_area.shp which contains all layer property values pre-populated. If you do this, you may skip the steps in this section.*

12. Create a new property polygon set to hold properties that have a constant value over the entire model domain:
    Right-click Property Polygon Sets and choose New Property Polygon Set.
    a. Left-click to select the new polygon set that appears at the bottom – it should be called Property Polygon Set 1.
    b. Go to the Edit menu and ensure Edit Property Polygons is the active edit mode.
    c. Holding down the Ctrl key, left-click just outside the top-left corner of the model boundary to create the first vertex of a new boundary polygon.
    d. Holding down the Ctrl and Shift keys, left-click outside the top-right corner of the model boundary to add another vertex.
    e. Holding down the Ctrl and Shift keys, left-click outside the bottom-right corner of the model boundary to add another vertex.
    f. Holding down the Ctrl and Shift keys, left-click outside the bottom-left corner of the model boundary to add the final vertex to the polygon. You should end up with a rectangle covering the entire model.
       If you make a mistake and create a new, second polygon, you can select the vertex you created by left-clicking it, and then press the Del key to remove it. Alternatively, you can right-click the property polygon set layer and choose Remove All Polygons in Set, then start digitising the polygon again.



13. Rename the polygon set to **Entire model area**:
    a. Right-click Property Polygon Set 1 and choose Rename.
    b. Type Entire model area and hit Enter.
14. Add properties Kx1 to Kx7, Kz1 to Kz7, ET_Depth, ET_rate and Recharge to the property polygon set:
    a. Right click the Entire model area polygon set and choose New Property.
    b. Give the property the name Kx1 and a default value of 5 (from Layer 1's Kx entry on the first page of this tutorial).
    c. Repeat steps a-b to add the following properties with their associated values:

| | | | |
|---|---|---|---|
| **Kx2** (0.1) | **Kx3** (1e-5) | **Kx4** (1e-4) | **Kx5** (1e-2) |
| **Kx6** (1e-4) | **Kx7** (1e-5) | | |

| | | | |
|---|---|---|---|
| **Kz1** (0.5) | **Kz2** (0.01) | **Kz3** (1e-6) | **Kz4** (1e-5) |
| **Kz5** (1e-3) | **Kz6** (1e-5) | **Kz7** (1e-6) | |

**ET_Rate** (3e-4)
**ET_Depth** (2)
**Recharge** (4e-4)

## Map Hydraulic Conductivities (Quick Start: tut2-stage3.amproj)

15. Under the Entire model area polygon set, map Kx1 through Kx7 to Kx and Kz1 through Kz7 to Kz for their corresponding layers:
    a. Right-click Kx1 and choose Map to Mesh Variable->Kx.

b. Ensure the new mapping is set to Layer 1.
c. Right-click Kx2 and choose Map to Mesh Variable->Kx.
d. Right-click the new mapping and choose Edit Layers.
e. Type the number 2 and hit Enter to change the mapping to Kx (Layer 2).
*f.* Repeat steps c-e for Kx3 through Kx7 (mapping to layers 3 to 7) and Kz1 through Kz7 (mapping to mesh variable Kz in layers 1 to 7).
*Alternatively, you may find it quicker to edit a .PPM file containing these mappings in a text editor, instead of mapping each of the properties manually. To do this, first map just Kx1, Kx2, Kz1 and Kz2. Then right-click on Property Polygon Sets and choose Save Property Polygon Mapping (.PPM) File. Open the saved PPM file in a text editor and navigate to the bottom of the file. Continue the pattern you see for the layer 1 and 2 mappings of Kx and Kz to create the mappings for the rest of the layers. The resulting mapping section of the file should look something like this:*

```
BEGIN PROPERTY_POLYGON_MAPPING
    SP 1
        LAYER 1
            MESH_CELL_EDGE_LENGTH PROPERTY_POLYGON_SET_3 CellSize
            NO_REFINE PROPERTY_POLYGON_SET_2 NO_REFINE
            MESH_CELL_EDGE_LENGTH PROPERTY_POLYGON_SET_1 CellSize
            KX PROPERTY_POLYGON_SET_4 Kx1
            KZ PROPERTY_POLYGON_SET_4 Kz1
        LAYER 2
            KX PROPERTY_POLYGON_SET_4 Kx2
            KZ PROPERTY_POLYGON_SET_4 Kz2
        LAYER 3
            KX PROPERTY_POLYGON_SET_4 Kx3
            KZ PROPERTY_POLYGON_SET_4 Kz3
        LAYER 4
            KX PROPERTY_POLYGON_SET_4 Kx4
            KZ PROPERTY_POLYGON_SET_4 Kz4
        LAYER 5
            KX PROPERTY_POLYGON_SET_4 Kx5
            KZ PROPERTY_POLYGON_SET_4 Kz5
        LAYER 6
            KX PROPERTY_POLYGON_SET_4 Kx6
            KZ PROPERTY_POLYGON_SET_4 Kz6
        LAYER 7
            KX PROPERTY_POLYGON_SET_4 Kx7
            KZ PROPERTY_POLYGON_SET_4 Kz7
END
```

*Once that is done, right-click on Property Polygon Sets and choose Remove All Property Polygon Sets. Then right-click Property Polygon Sets and choose Load Property Polygon Mapping (.PPM) File, and load the PPM file you have modified.*

16. With the entire model area properties mapped, you may want to hide the polygon so that the rest of the model and mesh is more easily visible. To do this, right-click the Entire model area layer and toggle the Visible flag off.
17. Under the Alluvium polygon set, map AlluviumKx to Kx (Layer 1) and AlluviumKz to Kz (Layer 1). This will apply altered Kx and Kz values within the alluvium polygon, as per the layer specifications on the first page of this tutorial.
18. In the list of polygon sets, ensure that the Alluvium layer appears above the Entire model area layer. This is important: if the alluvium layer is below the Entire model area layer, cells within the alluvium will instead take the regolith values from the Entire model area layer.

Whenever polygon layers are nested, one inside the other, like this, the mapping in the topmost layer in the list takes precedence.

## Add Boundary Conditions and Initial Head Mapping (Quick Start: tut2-stage4.amproj)

19. Right-click the WellQ property under the production_bore_polygons polygon set, and choose Map to Mesh Variable->Well Pumping Rate. You can inspect the pumping rate that has been applied to each bore by selecting the WellQ property and zooming in on one of the production bore polygons. Move the mouse inside the polygon and note the value shown in the bottom-right of the AlgoMesh window (WellQ: -1000). Note that a negative pumping rate, as in previous versions of MODFLOW, represents groundwater abstraction, whereas a positive pumping rate would represent injection. All production bores in this tutorial model are pumping water out of the aquifer rather than injecting water.
20. Change the Well Pumping Rate mapping to Layer 2 (ignore the Stress Period number for now – it should remain as SP 1).
21. Right-click Recharge under Entire model area, and choose Map to Mesh Variable->Recharge. Leave the mapping as Layer 1, SP 1.
22. Under Entire model area, map ET_Depth to ET Extinction Depth (Layer 1, SP 1) and ET_Rate to ET Maximum Flux (Layer 1, SP 1).
23. Under the elevations property TIN, right-click top01 and choose Map to Mesh Variable->ET Surface Elevation (leave mapping as Layer 1, SP 1). This will use the model top elevations as the evapotranspiration surface.
24. Also map top01 to Initial Head.
25. Right-click the new top01 Initial Head mapping, and choose Edit Layers.
26. Type **all** and hit Enter. This will apply the same Initial Head values to all seven layers.
27. Add the river property TIN and map the RIV boundary condition values to its properties:
    a. Right-click Property TINs and choose New Property TIN.
    b. Select river.mdd and click Open.
    c. Map Riverbed to River Bottom Elevation (Layer 1, SP 1).
    d. Map RiverK to River Conductance per Unit Area (Layer 1, SP 1).
    e. Map Stage to River Stage (Layer 1, SP 1).
28. Add the CHD polygon shapefile and map its CHD property to Constant Head (Layers 1-2):
    a. Right-click Property Polygon Sets and choose Import GIS Polygons.
    b. Select CHD_all_polys.shp and click Open.
    c. This polygon layer contains a significant number of polygons (29077), as it was created directly from cells in a raster in GIS. The large number of polygons may cause AlgoMesh to slow down on some systems. As such, you may want to make these polygons invisible by right-clicking the CHD_all_poly layer and toggling the Visible flag off.
    d. Map the CHD property to Constant Head.
    e. Right-click the Constant Head property mapping and choose Edit Layers.
    f. Type **1-2** and hit Enter to apply the mapping to layers 1 and 2.

## Export, Customise and Run the MODFLOW-USG Model (Quick Start: tut2-stage5.amproj)

29. We are now ready to export MODFLOW-USG model files from AlgoMesh. Go to File->Voronoi Mesh->Export to MODFLOW-USG.

30. Find the folder off the Tut2 folder called Working, enter this folder and provide a model name prefix for the USG model, e.g. **ss01**, and click Save.
31. Wait for AlgoMesh to finish writing the model files; this may take a minute or so.
32. You can now go into the Tut2\Working folder in Windows Explorer and inspect the model files as desired, using a text editor.
33. We need to modify the WEL package for this model. Open the file ss01.WEL in Notepad (or your favourite text editor).
34. Add the keyword AUTOFLOWREDUCE at the end of the second line of the file. The top of your WEL file should now look something like this:

```
# MODFLOW-USG Well (WEL) package exported by HydroAlgorithmics' AlgoMesh 1.0.5.
148 0 AUTOFLOWREDUCE
148 0 0
13818  -1000
13819  -1000
```

    This keyword ensures that the wells we have put in the model will reduce their pumping rate appropriately when there is no water in the cell they are pumping from.

35. We will also edit the LPF package for this model. Open the file ss01.LPF in your text editor.
36. Add the keywords NOVFC CONSTANTCV at the end of the second line of this file. The top of your LPF file should now look something like this:

```
# MODFLOW-USG Layer-Property Flow (LPF) package exported by HydroAlg
0 1e+030 0 0 NOVFC CONSTANTCV
 4 4 4 4 4 4 4  LAYTYP(NLAY) Layer type (confined/convertible)
 0 0 0 0 0 0 0  LAYAVG(NLAY) Hydraulic conductivity averaging scheme
 1.0 1.0 1.0 1.0 1.0 1.0 1.0  CHANI(NLAY) Anisotropy factor
 0 0 0 0 0 0 0  LAYVKA(NLAY) Vertical hydraulic conductivity flag
 0 0 0 0 0 0 0  LAYWET(NLAY) Wetting flag
```

    These keywords remove some nonlinearities from the vertical flow calculations which are not needed for our model. Running with these two keywords active produces equivalent vertical flow calculations to previous versions of MODFLOW (MODFLOW-NWT and earlier).

37. Now we are ready to run the model. Open a Command Prompt (e.g. Windows key+R, type **cmd**, and hit Enter; or go to the Start menu and search for Command Prompt).
38. Change to the full path of your Working folder, e.g. if your workshop files are under the folder c:\amdata\workshop, type:
    **cd /d "c:\amdata\workshop\Tut2\Working"**
39. Run the model:
    **mfusg.exe ss01.NAM**
    *or* **mfusg_x64.exe ss01.NAM** *to run the 64-bit version, if desired*
40. Wait for the model run to complete. How long this takes will depend on your computer, but it should complete within 5-10 minutes, reporting "Normal termination of simulation".
41. Open up the new ss01.LST file in Notepad or another text editor, and scroll to the bottom. Check the mass balance error; it should be very small (0.00% reported).

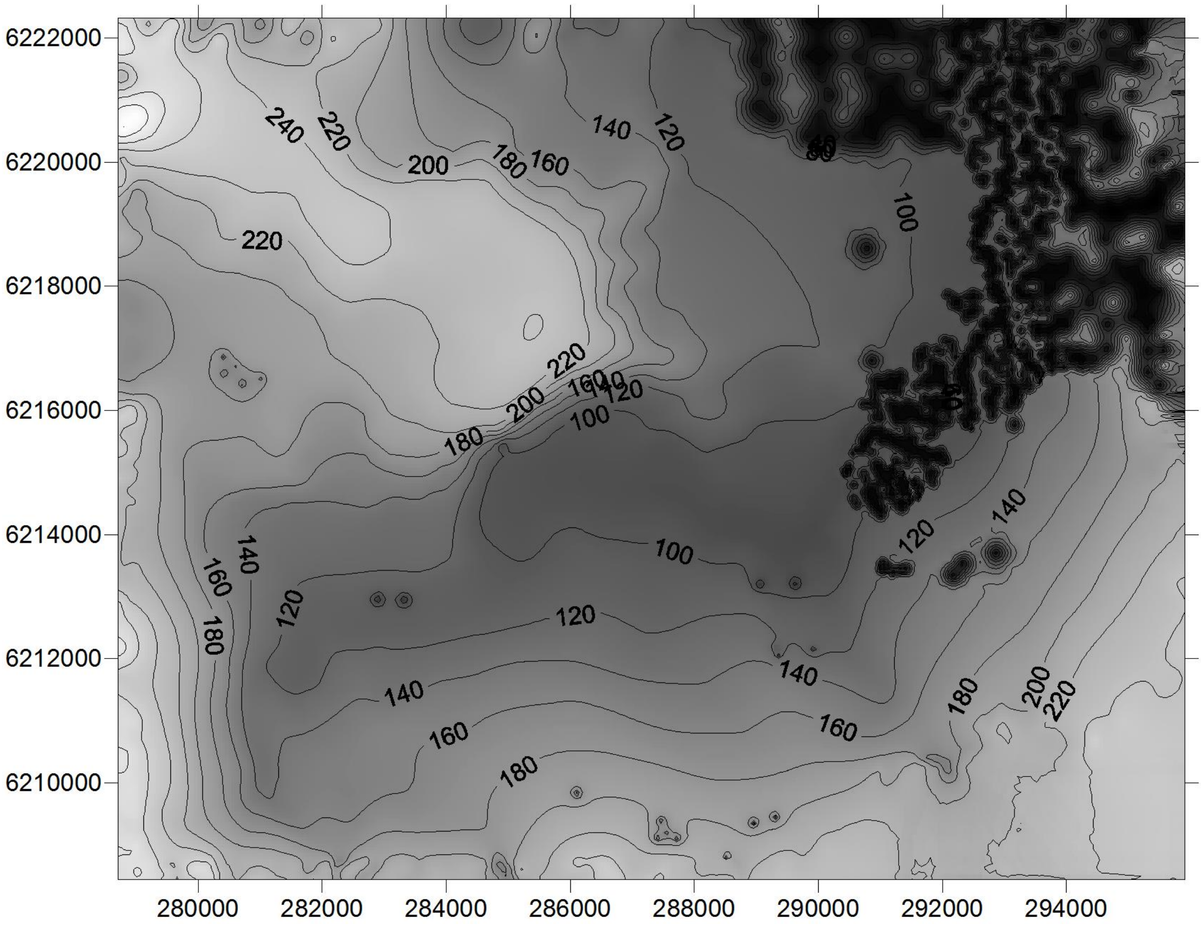
## Import Simulated Head Results

42. Now that the model run is complete, we would like to bring the steady-state simulated head results into AlgoMesh to inspect and to export for use with other post-processing tools. Go to File->Voronoi Mesh->Convert MODFLOW-USG Binary Head/Drawdown to MDD.
43. Select the file ss01.HDS and click Open.

44. When the Save As window appears, give a name for a new MDD file to be created as a property TIN containing all layer head values, e.g. ss01-heads.mdd.
45. When prompted if you would like to load the MDD now as a property TIN, click Yes.

### Export Simulated Heads to CSV (Quick Start: tut2-stage6.amproj)

46. Scroll down to the newly-added ss01-heads layer in the list of property TINs, and note that there is one property listed per layer in the model. If we had imported heads from a transient model, there would be one property per layer and per simulated time step.
47. Right-click HEAD_L2_SP1_TS1 and choose Voronoi Mesh->Export to CSV (Cell Centre Values).
48. Provide a name for the CSV file to be created, e.g. ss01-heads-l2.csv, and click Save.
49. You can now open the created CSV file in Excel, Surfer, or another tool of your choice to analyse the simulated head values.
    *Alternatively, you can skip the step of bringing heads into AlgoMesh and use File->Voronoi Mesh->Extract MODFLOW-USG Binary Head/Drawdown to CSV to extract a batch of multiple results entries directly to one or more CSV files.*



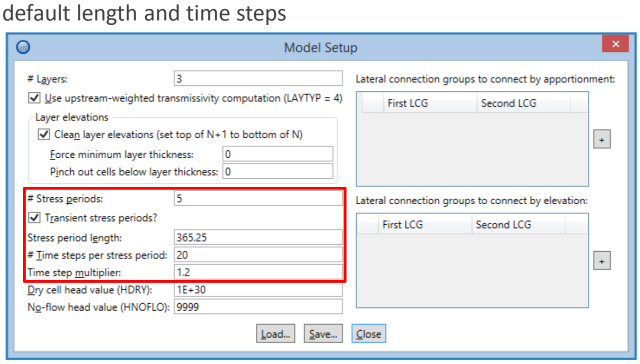

Minor drawdown cones can be seen at each of the production bores, and to some extent the shape of the river is also evident. The patchy effect in the north-eastern corner is due to layer 2 pinching out for some cells in the area; the interpolated head surface has not been clipped to show this here. To improve on this, you would need to filter out head values belonging to cells that are not present in layer 2 of the model – using perhaps the layer to global cell index CSV exported by AlgoMesh.

# Transient Model Setup

SESSION 4: MODFLOW-USG MODEL BUILDING (TRANSIENT) AND CONCLUSION

# Stress Period Setup

- Need to modify default Model Setup parameters for a transient model
  - Specify number of stress periods (SPs)
  - Set stress period type to transient
  - Set default length and time steps



HydroAlgorithmics

# Stress Period Setup

- Need to edit stress period schedule at end of DISU file after AlgoMesh has exported MODFLOW-USG files
  - Change length and time step parameters of any differing stress periods

```
0    5225.940824 1238.936833 2063.481351 1072.677017 1810.925355 402.259667  2207.805490 2015.051108
0    1876.476530 861.921556  2159.173907 402.259667  1386.434428
0    2081.781898 1136.186519 2207.805490 476.228415  1176.445313
0    2163.694073 476.228415  2015.051108 1386.434428 1250.000000
0    919.097900  1176.445313 1250.000000
365.25 20 1.2 TR
365.25 20 1.2 TR
365.25 20 1.2 TR
365.25 20 1.2 TR
365.25 20 1.2 TR
```

- May also need to edit OC file if number of time steps is changed

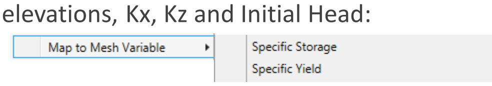
```
HEAD SAVE UNIT 30
HEAD PRINT FORMAT 0
DRAWDOWN SAVE UNIT 31
DRAWDOWN PRINT FORMAT 0
PERIOD 1  STEP 1
    SAVE HEAD
    SAVE DRAWDOWN
    SAVE BUDGET
    PRINT BUDGET
PERIOD 1  STEP 2
    SAVE HEAD
    SAVE DRAWDOWN
    SAVE BUDGET
    PRINT BUDGET
```



HydroAlgorithmics

# Transient Model Properties

- A transient model must have the following properties mapped for all layers, in addition to elevations, Kx, Kz and Initial Head:
  - Specific Storage (Ss)
  - Specific Yield (Sy)




- Any cells with missing values will take a 0 value
  - Be careful to map entire model domain for all layers
- The same three mapping mechanisms are supported, as before:
  - Property polygon sets
  - Property tables
  - Property TINs

# Transient Boundary Conditions

- By default, boundary conditions apply only to SP 1
- Right-click mapping -> Edit Stress Periods to change
  - Flexible textual entry of stress periods
    - Single SP, e.g. **2**
    - Range of SPs, e.g. **1-5**
    - Combinations of single SPs and ranges of SPs, e.g. **2,3-7,8,10,12-15**
    - All SPs: **All** or *
- BC will be applied only at specified SPs, and turned off at others
  - Exceptions: RCH and EVT
    (if active, you must provide values at every stress period)



# Cell-by-cell Budget Files

- MODFLOW-USG can output binary cell-by-cell budgets for various BC packages, like traditional MODFLOW
- AlgoMesh does not enable this functionality by default;
  you must manually change USG model files to enable it

1. Add BINARY budget output file line to NAM, assign unit number

```
DATA(BINARY) 50 C:\Consulting\USG\AMValidation\Working\amv6tr.CBB
```

2. Add unit number in appropriate place in desired BC files

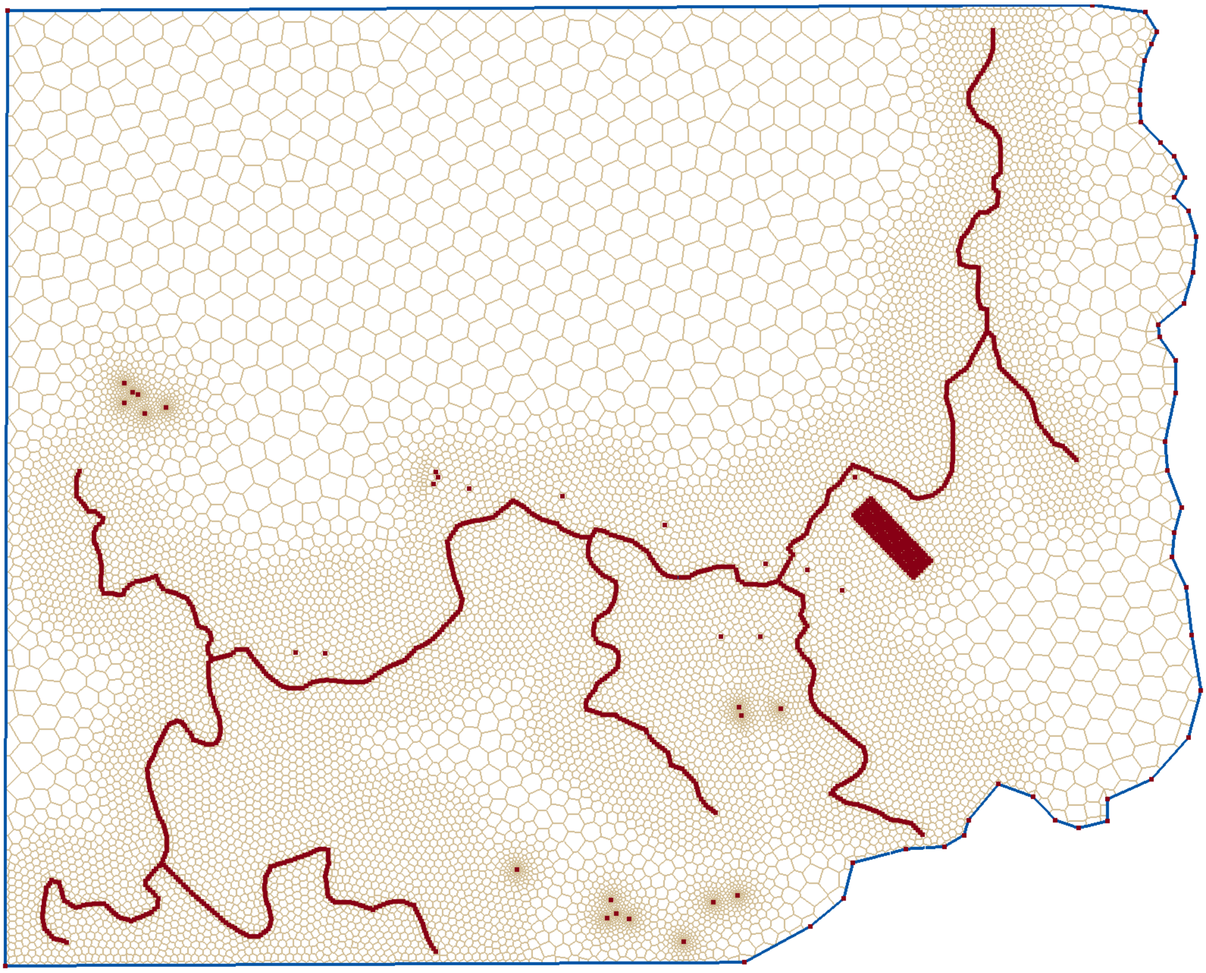
```
# MODFLOW-USG Drain (DRN) package exported by HydroAlgorithmics' AlgoMesh 1.0.5.
1280 50
0 0
```

3. Add SAVE BUDGET at desired time steps in OC file

```
PERIOD 1  STEP 1
    SAVE HEAD
    SAVE DRAWDOWN
    SAVE BUDGET
    PRINT BUDGET
```

## Tutorial 3: Transient Model Building

In this tutorial you will take the steady-state MODFLOW-USG model built in Tutorial 2 and convert it into a transient model, with the addition of drains simulating an underground mining activity.



### Model Layers

As in Tutorial 2, the model contains seven layers, detailed below.

| Layer | Description | Indicative Thickness | Kx | Kz | Ss | Sy |
|---|---|---|---|---|---|---|
| 1 | Regolith / Alluvium | 50m | 5 / 10 | 0.5 / 1 | 1e-5 / 1e-5 | 0.1 / 0.2 |
| 2 | Aquifer | 100m | 0.1 | 0.01 | 1e-5 | 0.1 |
| 3 | Aquitard | 25m | 1e-5 | 1e-6 | 1e-5 | 0.005 |
| 4 | Aquifer | 150m | 1e-4 | 1e-5 | 1e-5 | 0.02 |
| 5 | Coal Seam | 5m | 1e-2 | 1e-3 | 1e-6 | 0.01 |
| 6 | Aquifer | 25m | 1e-4 | 1e-5 | 1e-6 | 0.02 |
| 7 | Aquitard | 100m | 1e-5 | 1e-6 | 1e-6 | 0.005 |

### Stress Period Schedule

| Stress Period | Length | Description |
|---|---|---|
| **1** | 1 year | Pre-mining |
| **2** | 1 year | Mining of longwall section 1 |
| **3** | 1 year | Mining of longwall section 2 |
| **4** | 1 year | Mining of longwall section 3 |
| **5** | 10 years | Post-mining recovery |

### Boundary Conditions

Existing WEL, RIV, CHD, RCH and EVT boundary conditions will be retained from the steady-state model and extended to run over the entire transient simulation.

DRN boundary conditions will be added to simulate mining of the longwall in the eastern part of the model, divided into three sections. The drains will be applied to layer 5 (the coal seam), with an elevation below the bottom of the layer in the longwall area (in this case a constant Drain Elevation mapping of -1). Note that in a serious model, this may be varied spatially by using either a property TIN or a direct cell-by-cell mapping from a CSV table. A constant conductivity of 1 m/day will be mapped to Drain Conductance per Unit Area. Each drain will be active from the start of its respective mining period (see the stress period schedule above) until the end of stress period 4. All drains will remain inactive for the entirety of stress period 5.

### Input Data Files

The following files are needed to set up material property and boundary condition mappings.

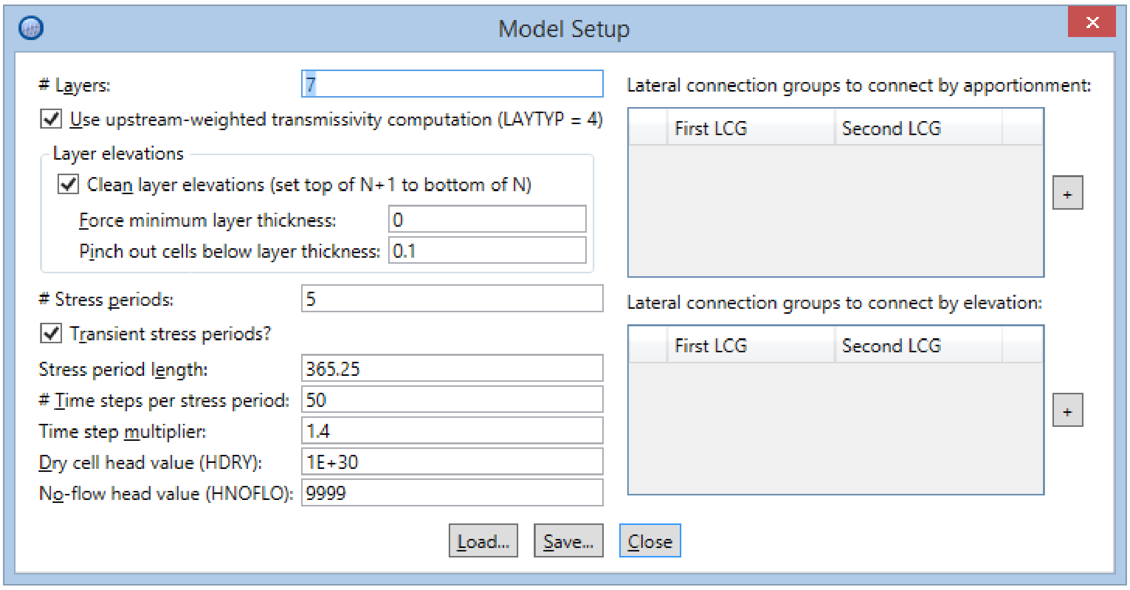
| | |
|---|---|
| **Tut3-start.amproj** | AlgoMesh project file containing the starting point for this tutorial: a complete steady-state model produced in Tutorial 2. |
| **Longwall1.shp** | Polygon shapefile outlining section 1 of the longwall to be mined. Contains drain elevation and conductance properties. |
| **Longwall2.shp** | Polygon shapefile outlining section 1 of the longwall to be mined. Contains drain elevation and conductance properties. |
| **Longwall3.shp** | Polygon shapefile outlining section 1 of the longwall to be mined. Contains drain elevation and conductance properties. |

## Step-by-Step Instructions

### Model Setup

1. Start AlgoMesh.
2. Go to File->Open Project, select tut3-start.amproj from the Tut2 folder and click Open. *Alternatively, you may continue from your own project if you have completed Tutorial 2.*
3. Once the model has opened and the mesh is displayed, go to Model->Model Setup.
4. Set # Stress Periods to 5.
5. Turn on Transient stress periods.
6. Set Stress period length to 365.25. We will later extend the length of the final recovery stress period, SP 5, by manually editing the DISU file.
7. Set # Time steps per stress period to 50.

8. Set Time step multiplier to 1.4. Your Model Setup window should look like this:




9. Close the Model Setup window.

## Extend Existing Boundary Conditions (Quick Start: tut2-stage1.amproj)

10. We now need to make sure our existing boundary conditions from the steady-state model will be active for the entire simulation. Start by right-clicking the Well Pumping Rate mapping under the WellQ property of the production_bore_polygons layer and choosing Edit Stress Periods.
11. Type **all** and hit Enter to activate the well boundary condition for all stress periods.
12. Repeat steps 9-10 to set All Stress Periods for the mappings to all of the following variables:

    ET Extinction Depth
    ET Maximum Flux
    ET Surface Elevation
    Recharge
    Constant Head
    River Bottom Elevation
    River Conductance per Unit Area
    River Stage

## Set Initial Head from Simulated Steady-State Results (Quick Start: tut2-stage2.amproj)

13. Scroll down to the ss01-heads property TIN, which contains the imported heads from the steady-state simulation.
14. Map HEAD_L1_SP1_TS1 to Initial Head (Layer 1).
15. Repeat step 13 to map the remaining six head properties to their corresponding layers:
    HEAD_L2_SP1_TS1 to Initial Head (Layer 2)
    HEAD_L3_SP1_TS1 to Initial Head (Layer 3)
    HEAD_L4_SP1_TS1 to Initial Head (Layer 4)
    HEAD_L5_SP1_TS1 to Initial Head (Layer 5)
    HEAD_L6_SP1_TS1 to Initial Head (Layer 6)
    HEAD_L7_SP1_TS1 to Initial Head (Layer 7)

16. Notice that the original steady-state Initial Head mapping under the elevations->top01 property has now changed to Initial Head (No Layers). This is because you can only have a single property TIN mapped to each layer of this property. As you added Initial Head mappings to each layer from the simulated head results, AlgoMesh automatically removed these layers from the top01 mapping. This old mapping may now be safely removed if desired, by right-clicking the Initial Head (No Layers) entry and choosing Remove Mapping.

### Map Storage Properties (Quick Start: tut2-stage3.amproj)

17. Under the Entire model area polygon set, map the fields Ss1 through Ss7 to Specific Storage in layers 1 through 7 respectively, and Sy1 through Sy7 to Specific Yield in their respective layers.
    *Note: if you manually created this layer in Tutorial 2 instead of importing it, you will need to add these properties manually, and set their default values according to the layer table on the first page of this tutorial.*
18. Under the Alluvium polygon set, map the field AlluviumSs to Specific Storage (Layer 1) and map the field AlluviumSy to Specific Yield (Layer 1). These properties will override the entire model area values in layer 1 for cells inside the alluvium polygon.

### Add Drain Boundary Conditions (Quick Start: tut2-stage4.amproj)

19. Right-click Property Polygon Sets and choose Import GIS Polygons.
20. Select Longwall1.shp and click Open.
21. Repeat steps 19-20 to bring in Longwall2.shp and Longwall3.shp.
22. Under Longwall1, map DrainElev to Drain Elevation.
23. Right-click the Drain Elevation mapping and choose Edit Layers.
24. Type **5** to apply the mapping to layer 5.
25. Right-click the Drain Elevation mapping and choose Edit Stress Periods.
26. Type **2-4** to apply the mapping from stress periods 2 to 4, inclusive.
27. Under Longwall1, map DrainK to Drain Conductance per Unit Area.
28. Change the Drain Conductance per Unit Area mapping to apply to layer 5.
29. Change the Drain Conductance per Unit Area mapping to apply to stress periods 2 to 4.
30. Repeat steps 22-29 to establish the remaining drain mappings as follows:
    Longwall2: DrainElev to Drain Elevation (Layer 5, SP 3-4)
    Longwall2: DrainK to Drain Conductance per Unit Area (Layer 5, SP 3-4)
    Longwall3: DrainElev to Drain Elevation (Layer 5, SP 4)
    Longwall3: DrainK to Drain Conductance per Unit Area (Layer 5, SP 4)

### Export and Run the Transient MODFLOW-USG Model (Quick Start: tut2-stage5.amproj)

31. Go to File->Voronoi Mesh->Export to MODFLOW-USG.
32. Enter the Working folder off the Tut3 folder, specify a model name prefix, e.g. tr01, and click Save.
33. Wait for AlgoMesh to finish writing the model files. This may take a minute or two.
34. As in Tutorial 2, we first need to modify the tr01.WEL file to activate the AUTOFLOWREDUCE keyword:
    a. Load tr01.WEL into Notepad or another text editor.
    b. Add the word AUTOFLOWREDUCE at the end of the second line of the file, as was done for the steady-state model.

35. Also as in Tutorial 2, we will need to modify the tr01.LPF file to active the NOVFC and CONSTANTCV keywords:
    a. Load tr01.LPF into Notepad or another text editor.
    b. Add the words NOVFC CONSTANTCV at the end of the second line of the file, as was done for the steady-state model.
36. We also need to change the length of the recovery period, stress period 5, as AlgoMesh always outputs the same length for each stress period:
    a. Open the file tr01.DISU in a text editor.
    b. Scroll to the very bottom of the file. You should see five lines at the end, specifying the length, number of time steps, time step multiplier and stress period type for each stress period:

```
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR

```

    *c.* Change the length of the last stress period to 3652.5 days (10 years), so that the end of your DISU file looks like this:

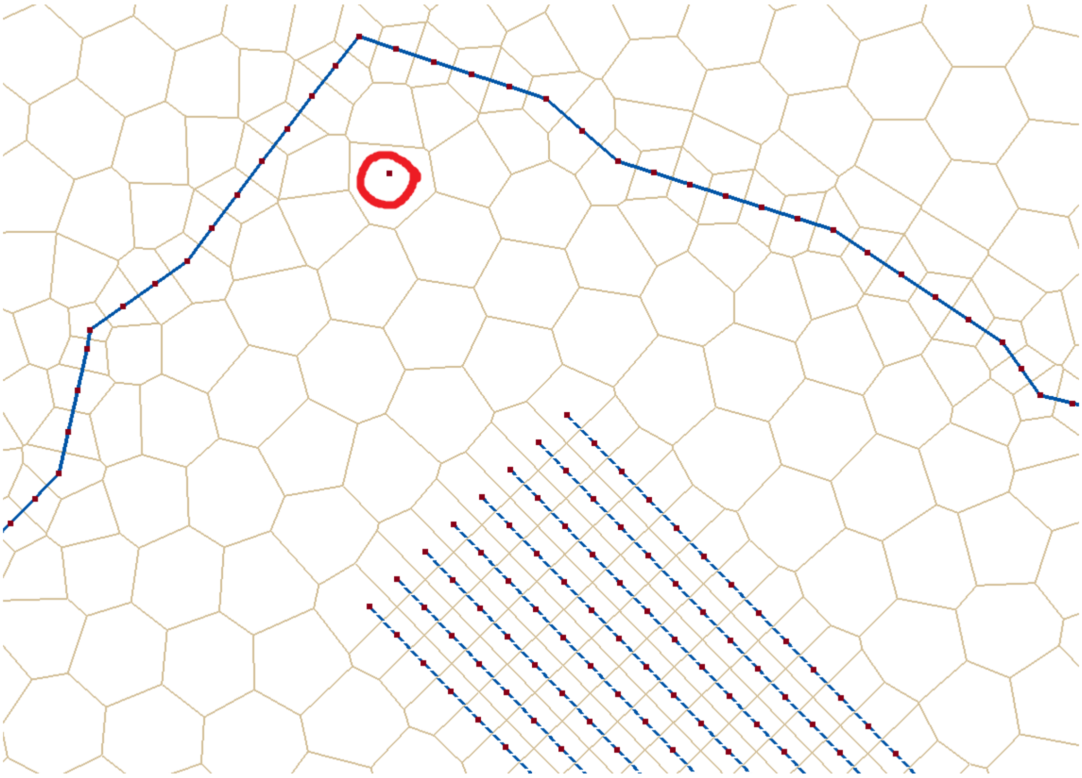
```
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
365.250000 50 1.400000 TR
3652.50000 50 1.400000 TR

```

    *Note: if needed, you can also increase the number of time steps in the final stress period here. However, if you do so, you will also need to modify the tr01.OC file and add an entry for each additional time step to ensure that head data is saved and budget data is printed for the new time steps.*
37. We are now ready to run the transient model. Open a Command Prompt (e.g. Windows key+R, type **cmd**, and hit Enter; or go to the Start menu and search for Command Prompt).
38. Change to the full path of your Working folder, e.g. if your workshop files are under the folder c:\amdata\workshop, type:
    **cd /d "c:\amdata\workshop\Tut3\Working"**
*39.* Run the model:
    **mfusg.exe tr01.NAM**
    *or* **mfusg_x64.exe tr01.NAM** *to run the 64-bit version, if desired*
40. Wait for the model run to complete. How long this takes will depend on your computer – it may take 10-20 minutes – but you should see the simulation progress through each of the time steps, reporting "Normal termination of simulation" at the end if all went smoothly.
41. Open up the new tr01.LST file in Notepad or another text editor, and scroll to the bottom. Browse through the file and inspect the listed mass balance errors at a few time steps to check that they look reasonable.

### Extracting Time-Series Results

42. As in Tutorial 2, you can extract CSV files from the transient .HDS file produced by the model if desired. Here, however, we are going to produce a hydrograph at an observation bore location instead. First, zoom in and identify the bore below in your AlgoMesh view:



The bore is just north of the mined longwall, close to a bend in the river.

43. Move your mouse over the centre of the cell, and note down the Voronoi cell number AlgoMesh reports in its bottom status bar:


In this case, it is 2D mesh cell index 4424.

44. We now need to identify the global cell index that this cell maps to for each of the seven layers in the model. Open up the tr01-layer-to-global-index.CSV file in a text editor.
45. Search for the text **1,4424,** - this will find the row in the CSV file giving the cell's global index in layer 1.

```
4425   1,4424,4424
```

As we can see, the global cell index of this cell in layer 1 is the same as the 2D mesh cell index. This is because our model does not remove any cells in layer 1.

46. Repeat step 45 for layer 2 (search for **2,4424,**) and layers 3 through 7, and note down the global cell index of the cell in each layer.

```
18138  2,4424,-1
31851  3,4424,30577
45564  4,4424,44012
59277  5,4424,57335
72990  6,4424,70926
86703  7,4424,84606
```

As the global cell index is listed as -1 in layer 2, the cell is not present in this layer. All other layers contain a cell at this location however, so we will extract hydrographs for each of these.

47. In a text editor, create a new file called tr01.n2b. Write a line for our observation bore in each of the layers 1 and 3 through 7, giving the global cell index of each in the format below:

```
Bore1_L1     1    4424     1
Bore1_L3     1    30577    1
Bore1_L4     1    44012    1
Bore1_L5     1    57335    1
Bore1_L6     1    70926    1
Bore1_L7     1    84606    1
```

This format is used by the USGMOD2OBS utility, which comes with the PEST Groundwater Data Utilities suite. We will use this utility to extract time series head values throughout the simulation at each of the cells listed in the tr01.n2b file.

48. In the text editor, open up the provided file usgm2s.in. Its contents should be as follows:

```
tr01.n2b
tr01.n2b
g
n
tr01.hds
250
998.9
d
30/06/1980
00:00:00
tr01.smp
```

This is a “piped” input file that will be used to answer the questions asked by the USGMOD2OBS utility. It has been set up in this case for this model – with 250 time steps and heads stored in the file tr01.hds. If you have changed any of these details for your model, modify the appropriate entries in this file accordingly. The tr01.smp entry gives the name of the output file that the utility should produce.

49. In your command prompt, run the following:
**usgmod2obs.exe < usgm2s.in**
*(The ‘<’ instructs Windows to use usgm2s.in as a piped input file; that is, the lines in that file are used in place of regular keyboard input to the utility, so that we don’t have to manually enter the data every time we run the utility)*

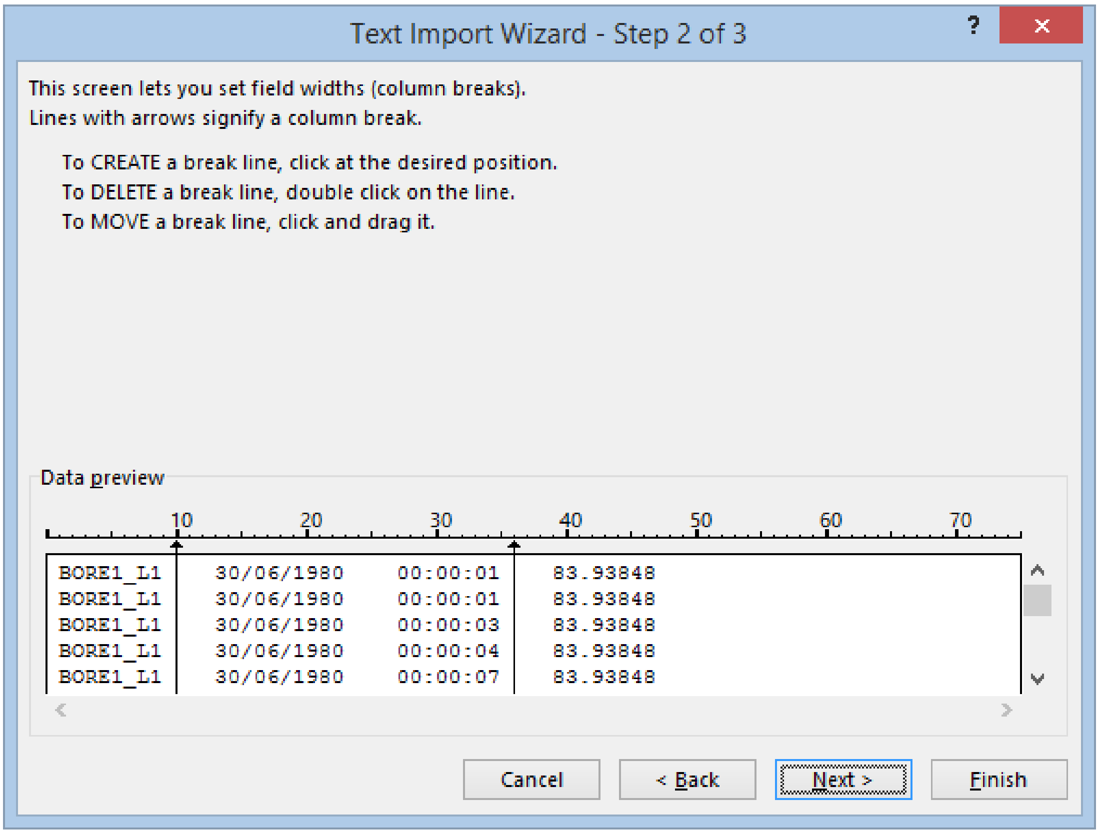
50. If the utility ran successfully, you should see the following at the end of its output:

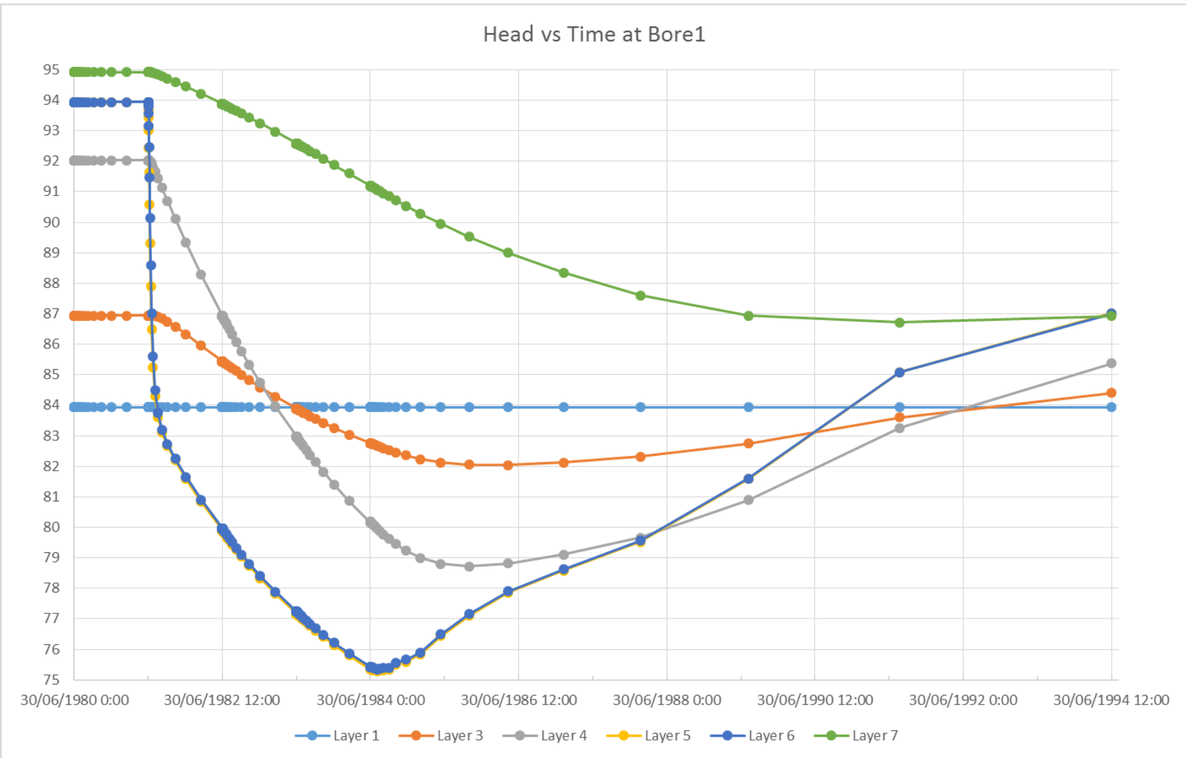
```
- reading unformatted MFUSG-generated file tr01.hds...

- 1750 arrays, covering 250 different model output times, read from file
  tr01.hds
- bore sample file tr01.smp written ok.
```

51. Opening up the produced tr01.smp file in a text editor shows that a series of time versus head entries for each of Bore1_L1, Bore1_L2, etc. have been generated from the transient simulation results. This file can be imported into Microsoft Excel, or other software, to plot the results as hydrographs.
*Hint: when loading the file into Excel, specify Fixed Width for the file type, and remove the break between the date and time columns before importing. This will ensure the time series*

*is kept in order, with a single date-time field in column 2.*



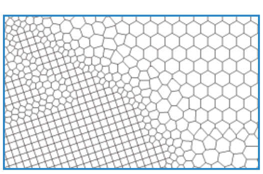



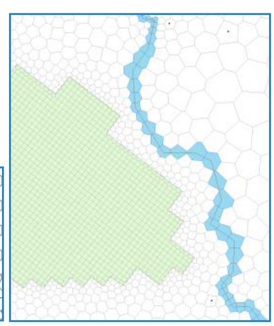
# Conclusion

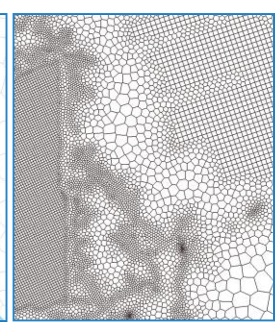
SESSION 4: MODFLOW-USG MODEL BUILDING (TRANSIENT) AND CONCLUSION

# Advanced Features

- AlgoMesh has several advanced features that haven't been covered in this workshop:
  - Structured sub-grids

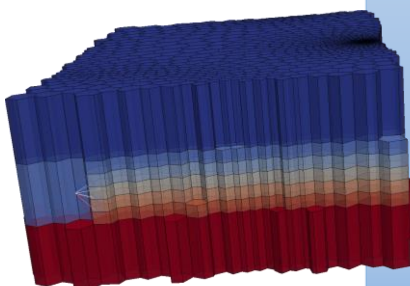


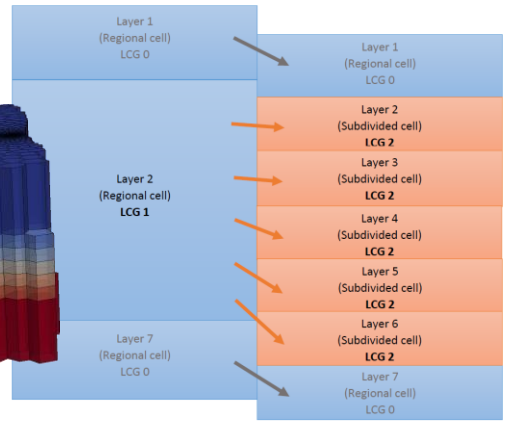




# Advanced Features

- AlgoMesh has several advanced features that haven't been covered in this workshop:
  - Structured sub-grids
  - Vertical sub-discretisation

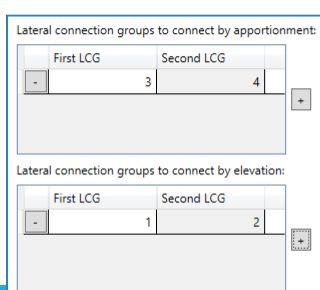


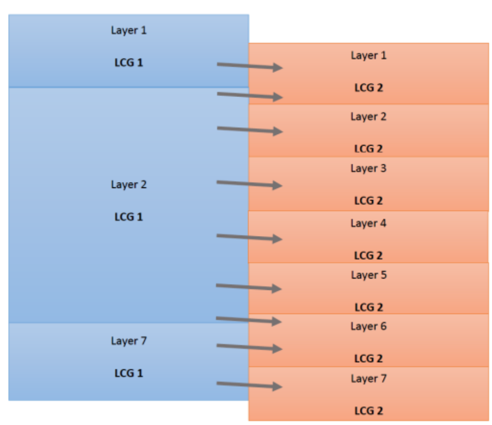





# Advanced Features

- AlgoMesh has several advanced features that haven't been covered in this workshop:
  - Structured sub-grids
  - Vertical sub-discretisation
  - Lateral connection by elevation across vertical faults



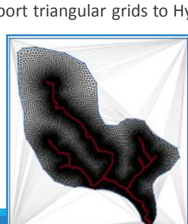
## Advanced Features

- AlgoMesh has several advanced features that haven't been covered in this workshop:
  - Structured sub-grids
  - Vertical sub-discretisation
  - Lateral connection by elevation across vertical faults
  - Python scripting (to be improved)



## Advanced Features

- AlgoMesh has several advanced features that haven't been covered in this workshop:
  - Structured sub-grids
  - Vertical sub-discretisation
  - Lateral connection by elevation across vertical faults
  - Python scripting (to be improved)
  - Mesh generation for other models
    - Export triangular grids to HydroGeoSphere (.AH2 format) or FEFLOW (shapefile)

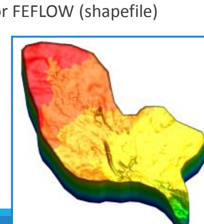


```
read algomesh 2d grid
mesh_test.ah2

generate layers interactive
    base elevation
        elevation constant
        600
    end ! base elevation

    new layer
       layer name
         topo
         elevation from raster file
        ./med_hat_filled.txt
    end

end !generate layers interactive
end !grid generation
```



# On the Wish List…

- Some possibilities we hope to explore in future versions of AlgoMesh:
  - Layer-dependent discretisations
  - Colour fills and contouring
  - Automated mesh quality checks
  - (Semi-)automated geometry cleanup
  - Improved command-line tool for scripted workflows
  - Improved VTK export functionality
  - Better raster support
  - Faster optimisation (run on GPU?)
  - Improved support for other models (e.g. HydroGeoSphere)